Why TEX Should NOT Output PostScript — Yet: Addendum

Shane Dunne
University of Western Ontario

[Editor's note: The original article appeared in Volume 9, No. 1, pp. 37–39. The following addendum was received after the issue had been sent to the printer.]

Insert at the end of the last full paragraph, page 38. An example of such a precise specification can be found in the documentation for the X11 window system under development at M.I.T. [3]

Additional Reference:

[3] Gettys, J., Newman, R., and Schiefler, R.W. *Xlib: C Language X Interface Protocol Version 11.* Massachusetts Institute of Technology, Laboratory of Computer Science, Cambridge, Massachusetts, 1987.

ASCII Preview with vuTEX

Warren Wolfe

The call for an ASCII previewer for TEX (Brown, vol.9, no.1, 1988) prompts me to report the release of just such a program. Created to reduce the *edit* – TEX – *print* cycle and the associated costs in time and laser produced output, the program has been tested and modified since January, 1988, and has proven to be an effective tool in our TEX treasury. As the output device is nonspecific, *i.e., any ASCII device*, we have abandoned the usual `dvigen` name format and have given the program the name vuTEX.

Features

vuTEX was developed from Rokicki's `dvigen` model in a naive attempt to satisfy our own immediate needs for a previewer. Thus, the design model varies from that proposed by Brown, but nonetheless is justified in the manner in which it satisfies our own design criteria:

1. Words, lines, paragraphs, and pages must appear as integral units as produced by TEX.
2. Characters from special fonts and symbols should be represented in a meaningful way by ASCII characters.
3. Alignments, equations, and tables should be reproducible.
4. The program should be fast and easy to use.
5. The output should be directed to any ASCII device or to a file for screen editor viewing.

The result is a stripped down and rebuilt `dvigen` model in which many of the parts have the same name and intended function, but the way the job is done has been altered dramatically. The limited character set and resolution of the simplest ASCII device were chosen as the limits on the resources available to vuTEX in its generic form, but it is adaptable to use any special abilities of particular devices, such as overstrike and reverse imaging. vuTEX offers the following features:

- Input is from a standard `dvi` file and font `tfm` files.
- Output is printable ASCII code which may be addressed to a file (for screen editor viewing) or to a device such as a terminal or lineprinter.
- Fonts of all sizes and most styles are simulated. Many special characters are mapped onto ASCII characters which hint at their true meaning, e.g. S for $\int$. Ligatures are represented by the group of characters so that ffl appears as `ffl`. Unrepresented characters are replaced with `#`.
- Used fonts are ranked so that questions of overlays or overstrikes are resolved by priority.
- The process is fast. Typical output on a Honeywell DPS8 is 1024 processed characters per second with a 60 page, 167,000 character, document processed to a file in less than 3 minutes.
- Horizontal spacing may be selected to retain the vertical alignment produced by TEX (for tables and equations) or to compress interword spacing (for ordinary text).
- Sub/superscripts may appear on different lines than the base line.
- Output may be truncated at the right margin. The coarse resolution of the fixed pitch font results in output that is wider than the 80 characters on a typical terminal screen. To avoid wraparound, the user can select the width of the printed output. Some terminals, and most printers, will print 132 characters to a line while many screen editors allow viewing of wide records with single key operations. Thus, vuTEX avoids the left-right-centre views proposed by Brown.