

## An ASCII Previewer for T<sub>E</sub>X

Marcus Brown  
Texas A&M University

A common complaint among users of T<sub>E</sub>X is the problem of the *edit-T<sub>E</sub>X-print* cycle, with the attendant delay of waiting on printer service. One solution has been to allow for viewing the formatted dvi file on the terminal. This has been effectively implemented on workstations. VorT<sub>E</sub>X for the SUN and a number of other previewers are all well documented. However, workstations are not universal for the T<sub>E</sub>X user.

Some earlier articles describe terminal previewers, but most have focused on graphics terminals, or on very low resolution representations on more common terminals such as the DEC VT100. These are generally not reasonable alternatives for the CRTs found on multi-user systems because of the limited I/O bandwidth. Most terminals are not capable of handling down-loaded fonts, and attempting a bit-map display frequently takes a minute or more to load a screen.

### Basic requirements

We believe that significant time and cost could be saved by developing a previewer for the normal ASCII terminal found in most working environments. In order to construct a usable previewer, we must consider what the user typically needs to see in a preview of his document:

1. The user will want to see any obvious errors, such as overfull hboxes, misspellings or typographic errors.
2. The user will want to check on line breaks, page breaks, and the placement of floating inserts and tables.
3. The user will want to check on any use of special fonts. For example, he will want to know if a closing brace has been omitted causing the last 5 pages to be set in *italic font*.
4. The user will want to see if the corrections made to problems detected in the last run gave the desired result.

Most of these types of questions can be satisfactorily answered using an ASCII previewer, and they constitute a large enough percentage of the trial prints of a document to justify the development and use of a previewer for an ASCII terminal.

### Design model

The following is a list of the relevant characteristics of ASCII terminals which we will use as a design model:

1. They have only a single fixed pitch font, although most have the ability to emphasize selected characters or regions of the screen by use of color, reverse-video, blinking, underlining, or brighter-than-normal characters.
2. They often have a limited graphics character set, but that is of little use in attempting to simulate typeset output such as math equations, for example. However, they usually allow for vertical and horizontal lines.
3. They normally have 24 vertical lines and 80 horizontal columns. This is obviously inadequate for viewing a typeset page, with 60 or more vertical lines and perhaps 100 horizontal characters plus margins on a typical page.

While this limited set of capabilities cannot do justice to a typeset document, it is able to provide enough information to the user to allow several types of corrections and adjustments. We propose that an ASCII terminal previewer should allow for the following options:

1. **Margins.** The previewer should allow viewing of the left and right margins of a page. The primary difficulty with this requirement is that the terminal does not have enough width to show a complete line, and in many cases the fixed width font obliterates T<sub>E</sub>X's efforts at right margin justification or alignment. With these two concerns in mind, the obvious solution is to allow viewing a subset of the page which shows the desired margins. We suggest three options for viewing margins. Each option is demonstrated using page 1 of the immortal *T<sub>E</sub>Xbook*.

- **Left Margin.** The lines shown on the screen should begin at the left margin of the screen. All letters typeset will be shown on the screen until the point at which the line must be truncated due to screen width. This method is demonstrated in Fig. 1. Note that even when all the letters fit on the line, the right margin is not justified.
- **Right Margin.** The right margin of the screen less 1 column should coincide with the right margin of the paper. The last column is reserved for the dreaded *overfull rule* which is simulated on the first line of Fig. 2. Characters should be written to the screen in reverse order from the right margin. The effect should be

---

This paper was supported in part by a contract with the U.S. Forestry Service.

that of truncating the left portion of each line where necessary and adjusting the alignment such that the final character on each line is at its proper position on the page.

- **Both Margins.** The left and right margins should be displayed as described above. Characters in the middle of the line will be deleted or added to allow both margins to appear justified on the screen. Fig. 3 should make this option clear.

These viewing alternatives will allow for checking line breaks, indentations and related information. The user should be able to switch back and forth between these alternatives with a single key stroke and a minimum of display time.

**2. Pages.** The previewer should allow the display of page breaks. This could be represented by displaying the top and bottom of a single page, with the middle lines omitted. It would also be desirable to allow display of the bottom of one page, a horizontal line representing the page break, and the top of the succeeding page. (This option should be added to more advanced previewers like VorTeX.) Perhaps the most commonly used option would allow for scrolling up and down through a single page and across page breaks to previous or succeeding pages. Again, the user should be able to switch easily between these options.

**3. Fonts.** The abilities of the terminals mentioned above to highlight certain characters could be used. Standard Roman fonts would be shown in the default text of the terminal, while color, reverse video or other options could be used to signify bold, slanted, etc. It would be unwise to attempt too complex a coding scheme here. Probably the best scheme would be to use only one or two alternative display techniques, with a *catch-all* category for all other fonts. For example, let `\bf` be signified by using brighter text and `\it` be signified with underlined text. Then all other fonts would be signified by reverse-video. If the representation of fonts becomes too complex, the user will spend more effort remembering the coding scheme than is justified by the additional information gained. These options might be adjustable by the user for different documents.

## Other issues

The display of tables should be done in a way that preserves the indentation of the left and right margins. If internal alignment is to be preserved when aligned columns are used, significant numbers of the characters in each column may be dropped, or some careful adjustment may be necessary when displaying the right margin. Horizontal and vertical rules surrounding the tables could be displayed using the limited graphics characters available.

The treatment of ligatures deserves consideration. One possibility might be to replace the ligatured characters with a number representing the number of characters included. Thus, 'ffi' would be replaced by '3', while 'fi' would be replaced by '2'. An alternative would be to replace the single ligature with the group of letters it represented. This would replace 'ffi' with 'ffl'. This has the disadvantage of taking up more screen space, but the previewer will already be dropping characters from the line, and it would be easier for the user to interpret.

Other special characters in a font and all characters from a primarily symbolic font would be represented by the default error symbol on most terminals, the checkerboard box, or perhaps by the '?' character, as in DVITYPE.

These specifications are designed to be achievable on the lowest common denominator of terminal likely to be in use in the TeX community. It is probable that most TeX users have terminals with some special capabilities which could be used to enhance this type of previewer. For example, we are currently implementing an ASCII previewer on a Data General 461, which allows for down-loadable fonts. These special fonts may be used for some special characters such as ligatures or other symbols.

While a TeX previewer for an ASCII terminal has obvious limitations, we believe that the timeliness of information conveyed to the user will justify the effort expended. We expect to have the previewer mentioned above available for distribution in March. It will also be adapted to run on a DEC VT220.

English words like 'technology' stem from a Greek root beginning with the letters `tech`...; and this same Greek word means art as well as technology. Hence the name `TEX`, which is an uppercase form of `tech`.

Insiders pronounce the `t` of `TEX` as a Greek chi, not as an 'x', so that `TEX` rhymes with the word `blecchhh`. It's the 'ch' sound in Scottish words like `loch` or German words like `ach`; it's a Spanish 'j' and a Russian 'kh'. When you say it correctly to your computer, the terminal may become slightly moist.

The purpose of this pronunciation exercise is to remind you that `TEX` is primarily concerned with high-quality technical manuscripts: Its emphasis is on art and technology, as in the underlying Greek word. If you merely want to produce a passably good document--something acceptable and basically readable but not really beautiful--a simpler system will usually suffice. With `TEX` the goal is to produce the finest quality; this requires more attention to detail but you will not find it much harder to go the extra distance, and you'll be able to take special pride in the finished product.

Figure 1: Display of Left Margin

h words like 'technology' stem from a Greek root beginning with the letters `tech`...; and this same Greek word means art as well as technology. Hence the name `TEX`, which is an uppercase form of `tech`.

Insiders pronounce the `t` of `TEX` as a Greek chi, not as an 'x', so that `TEX` rhymes with the word `blecchhh`. It's the 'ch' sound in Scottish words like `loch` or German words like `ach`; it's a Spanish 'j' and a Russian 'kh'. When you say it correctly to your computer, the terminal may become slightly moist.

The purpose of this pronunciation exercise is to remind you that `TEX` is primarily concerned with high-quality technical manuscripts: Its emphasis is on art and technology, as in the underlying Greek word. If you merely want to produce a passably good document--something acceptable and basically readable but not really beautiful--a simpler system will usually suffice. With `TEX` the goal is to produce the finest quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you'll be able to take special pride in the finished product.

Figure 2: Display of Right Margin

English words like 'technology' stem from a Greek root beginning with the letters `tech`...; and this same Greek word means art as well as technology. Hence the name `TEX`, which is an uppercase form of `tech`.

Insiders pronounce the `t` of `TEX` as a Greek chi, not as an 'x', so that `TEX` rhymes with the word `blecchhh`. It's the 'ch' sound in Scottish words like `loch` or German words like `ach`; it's a Spanish 'j' and a Russian 'kh'. When you say it correctly to your computer, the terminal may become slightly moist.

The purpose of this pronunciation exercise is to remind you that `TEX` is primarily concerned with high-quality technical manuscripts: Its emphasis is on art and technology, as in the underlying Greek word. If you merely want to produce a passably good document--something acceptable and basically readable but not really beautiful--a simpler system will usually suffice. With `TEX` the goal is to produce the finest quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you'll be able to take special pride in the finished product.

Figure 3: Display of Both Margins