

Leslie Lamport  
27 October 1988  
For L<sup>A</sup>T<sub>E</sub>X Version 2.09

## **Contents**

L<sup>A</sup>T<sub>E</sub>X runs on a variety of computers at many different sites. This document tells you how to use L<sup>A</sup>T<sub>E</sub>X on the Magic, Lewis, and Clark computers at SRC. It is not about L<sup>A</sup>T<sub>E</sub>X itself, which is described by the manual *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*, published by Addison-Wesley, available at fine book stores everywhere.

If you have a question that you can't answer by reading the manual and this document, ask Leslie Lamport. He should also be informed of any possible L<sup>A</sup>T<sub>E</sub>X bugs or undocumented anomalies.

## 1 Getting Started

### 1.1 Running a Sample File

Before preparing your own documents, you may want to get acquainted with L<sup>A</sup>T<sub>E</sub>X by running it on a sample input file. First make your own copy of the file `sample.tex` by typing the following Ultrix command:

```
cp /usr/local/lib/tex82/sample.tex .
```

(You must type the space followed by the period at the end. This and all other Ultrix commands are ended by typing *return*.) A copy of the file `sample.tex` is now in your current directory; you can edit it just like any other file. If you destroy or mess up your copy, typing the above command again gets you a fresh one.

Next, run L<sup>A</sup>T<sub>E</sub>X on the file `sample.tex` by typing:

```
latex sample
```

When L<sup>A</sup>T<sub>E</sub>X has finished, it will have produced the file `sample.dvi` in your directory. You can print this file by typing the command

```
iptex sample.dvi
```

The output will be produced on the Imagen printer, located next to the third floor receptionist's desk. If you want to print L<sup>A</sup>T<sub>E</sub>X output on an Apple LaserWriter, see Section 7.

After your output has been printed, you can delete `sample.dvi` by typing

```
rm sample.dvi
```

### 1.2 Preparing and Running L<sup>A</sup>T<sub>E</sub>X on Your Own Files

You must use a text editor to prepare an input file for L<sup>A</sup>T<sub>E</sub>X. The document *Welcome to SRC* describes the text editors available at SRC. The easiest way to start learning about L<sup>A</sup>T<sub>E</sub>X is by examining the file `small.tex` with your text editor. You can obtain your own copy of this file, in your directory, by typing the command

```
cp /usr/local/lib/tex82/small.tex .
```

After you have prepared your file, whose name should have the extension `tex`, you must run it through L<sup>A</sup>T<sub>E</sub>X and print the output. Follow the instructions in Section 2, except substitute the first name of your file for `ksample1`. Remember to save disk space by deleting the `dvi` file after printing the output.

If you want to stop L<sup>A</sup>T<sub>E</sub>X in the middle of its execution, perhaps because it is printing a seemingly unending string of uninformative error messages, type *Control-C* (press *C* while holding down the key labeled *CTRL*). This will make L<sup>A</sup>T<sub>E</sub>X stop as if it had encountered an ordinary error, and you can return to Ultrix command level by typing *X*, as described in the manual. If typing *Control-C* doesn't work, typing *Control-Z* will get you immediately to Ultrix command level, but this will leave a stopped job hanging around. A stopped job won't hurt anything and will disappear when you log out, but it forces you to type two successive `logout` commands to log out.

To use the *spell* program for finding spelling errors in a L<sup>A</sup>T<sub>E</sub>X input file named `myfile.tex`, type the following command:

```
delatex myfile.tex | spell
```

This will type a list of possibly misspelled words on your terminal. If you'd rather have the output written to a file named `foo.bar`, type

```
delatex myfile.tex | spell >foo.bar
```

## 2 Carrying On

### 2.1 L<sup>A</sup>T<sub>E</sub>X on Ultrix

The only special problems in using L<sup>A</sup>T<sub>E</sub>X caused by the Ultrix operating system involve the way Ultrix handles files. The first problem arises because, when a program starts to write a file, Ultrix destroys the previous version of that file. Thus, if an error forces you to stop L<sup>A</sup>T<sub>E</sub>X prematurely (by typing *Control-C* or *Control-Z*), then the files that L<sup>A</sup>T<sub>E</sub>X was writing are incomplete, and the previous complete versions have been destroyed. You probably don't care about the output on the `dvi` file, but, if you are making a table of contents or using cross-referencing commands, then L<sup>A</sup>T<sub>E</sub>X also writes one or more *auxiliary files* that it reads the next time it processes the same input file. If the auxiliary files are incomplete because L<sup>A</sup>T<sub>E</sub>X was stopped before reaching the end of its input file, then the table of contents and cross-references will be incorrect the next time L<sup>A</sup>T<sub>E</sub>X is run on the same input file. You will have to run L<sup>A</sup>T<sub>E</sub>X a second time to get them right. If you want to avoid having to run L<sup>A</sup>T<sub>E</sub>X twice after making an error—for example, if your input is very long—then you should save copies of these auxiliary files before running L<sup>A</sup>T<sub>E</sub>X. An input file named `myfile.tex` and all the auxiliary files produced by L<sup>A</sup>T<sub>E</sub>X from it are included in the Ultrix file specifier `myfile.*`. Use the Ultrix `cp` command to save copies of these files.

The second problem in using L<sup>A</sup>T<sub>E</sub>X on Ultrix involves the files that L<sup>A</sup>T<sub>E</sub>X reads. The file whose name you type with Ultrix's `latex` command is called the *root file*. In addition to reading the root file, L<sup>A</sup>T<sub>E</sub>X also reads the files specified by `\input` and `\include` commands. With the Ultrix directory system, L<sup>A</sup>T<sub>E</sub>X must know not only the names of these files but also on what directories they are. It will have no problem finding the correct files if you follow two simple rules:

1. Run L<sup>A</sup>T<sub>E</sub>X from the directory containing the root file.
2. Keep all files specified by `\input` and `\include` commands in the same directory as the root file.

If you follow these rules, you never have to type an Ultrix path specifier when using L<sup>A</sup>T<sub>E</sub>X.

You should never break the first rule, otherwise L<sup>A</sup>T<sub>E</sub>X will have trouble finding auxiliary files. (To run L<sup>A</sup>T<sub>E</sub>X on someone else's file, copy the file to your directory.) If you break the second rule, specifying a file from another directory in an `\input` or `\include` command, you must use a complete path name. For example, to include the file `hisfile.tex` from Jones's directory `/foo/bar`, you can type

```
\include/udir/jones/foo/bar/hisfile
```

A `~` character may not appear in the argument of an `\input` or `\include` command, so you *cannot* use a file name such as `~jones/foo/bar/hisfile`.

For people who don't like to obey rules, here is exactly how L<sup>A</sup>T<sub>E</sub>X finds its files. The root file is found by Ultrix according to its usual rules. L<sup>A</sup>T<sub>E</sub>X's auxiliary files are read and written in the directory from which it is run. All file names specified in the L<sup>A</sup>T<sub>E</sub>X input, including the names of document-style (`sty`) files specified by the `\documentstyle` command, are interpreted relative to the directory from which L<sup>A</sup>T<sub>E</sub>X is run. If L<sup>A</sup>T<sub>E</sub>X does not find a file starting in this directory, it looks in the system directory `/usr/local/lib/tex82`. You can change the directories in which L<sup>A</sup>T<sub>E</sub>X looks for its input files by setting the environment variable `TEXINPUTS`. Putting the command

```
setenv TEXINPUTS :./udir/jones/myown:/usr/local/lib/tex82:
```

in your `.login` file causes L<sup>A</sup>T<sub>E</sub>X to look for files first in the current directory, then in Jones's `/myown` directory, and then in the system directory. You might want to do this if your name is Jones and you have your own personal document-style files in your `/myown` directory.

## 2.2 Document Styles

There are nine document styles and style options available at SRC that are not described in the manual:

- the `proc` style option for making camera-ready copy for conference proceedings,

- The `bezier` option for drawing curves.

- The `ifthen` option for implementing **if-then-else** and **while-do** control structures.

- The `srcletter` style for making letters.

- The `showidx` option for printing index entries in the margin.

- The `makeidx` option for use with the *MakeIndex* program. (Section ? for information about this program).

- The `ps` style option that uses Times Roman fonts.

- The `preview` option for use with the *Proof*

The `ps-slides` style for use with `SLi T -.1667emEX` to produce color slides on the QMS color printer.

The `ps` option is described in Section ?; the remaining style options are described below.

### 2.2.1 The `proc` Style Option

The `proc` option is used with the `article` document style. It produces two-column output for ACM and IEEE conference proceedings. The command `\copyrightspace` makes the blank space at the bottom of the first column of the first page, where the proceedings editor will insert a copyright notice. This command works by producing a blank footnote, so it is placed in the text of the first column. It must go after any `\footnote` command that generates a footnote in that column.

L<sup>A</sup>T<sub>E</sub>X automatically numbers the output pages. It's a good idea to identify the paper on each page of output. Placing the command

```
\markrightJones---Foo
```

in the preamble (before the `\begin{document}` command) prints `kJonesnFool` at the bottom of each page.

### 2.2.2 The `bezier` Style Option

This option defines a single command, `\bezier`, that draws a curved line in a `picture` environment. Let  $p_i$  be the point with coordinates  $(x_i, y_i)$ , for  $i=1, 2, \text{ and } 3$ . The command

```
\beziern(x_1,y_1)(x_2,y_2)(x_3,y_3)
```

draws  $n$  points on the quadratic Bezier spline determined by the three points  $p_1, p_2, \text{ and } p_3$ . The locus of points on this spline is a parabolic arc from  $p_1$  to  $p_3$  having the line tangent to it at  $p_2$  and the line tangent to it at  $p_1$ . Note that  $p_2$  is *not* on this arc unless  $p_1, p_2, \text{ and } p_3$  are colinear, in which case the arc is a straight line. Bezier splines are useful because it's easy to join two of them together smoothly by giving them the same tangent line where they meet.

It takes roughly 75 points per inch to form a solid line, depending upon the line thickness. See Section C.13.3 of the manual for commands to specify line thickness in a `picture` environment. This command is *very* slow, and T<sub>E</sub>X has enough memory to hold only about 1000 points plus a page of text. (Remember that T<sub>E</sub>X keeps the current page plus all as yet unprinted figures in memory.) So, the `bezier` command should be used for only a small number of small curves.

### 2.2.3 The `ifthen` Style Option

This option provides two programming language features that are useful only for people who already know how to program. It defines the two commands

```
\ifthenelse{test}{then clause}{else clause}
```

```
\whiledo{test}do clause
```

that implement the following two Pascal language structures

<b>if</b> <i>test</i>	<b>then</b>	<i>then clause</i>
	<b>else</b>	<i>else clause</i>
<b>while</b> <i>test</i> <b>do</b>		<i>do clause</i>

The *then*, *else*, and *do* clauses are ordinary L<sup>A</sup>T<sub>E</sub>X input; *test* is one of the following:

A relation between two numbers formed with `<`, `>`, or `=`; for example, `\valuepage>3`.

`\equalstring1string2`, which evaluates to *true* if *string1* and *string2* are the same strings of characters after all commands have been replaced by their definitions. (Upper- and lowercase letters are unequal.)

A logical combination of the above two kinds of tests using the operators `\or`, `\and`, and `\not` and the parentheses `\(` and `\)` for example:

```
\not \( \valuesection = 1 \and \equalJones\myname \)
```

The *test* argument is a violently moving argument, which means that not only fragile commands but even some commands that are not normally fragile will break, causing T<sub>E</sub>X to enter an infinite loop. The `\protect` command works in these situations.

These commands, together with `\renewcommand` and the commands of Section C.7.4 for manipulating counters, open up a whole new world of hacking.

## 2.2.4 Letters

The `letter` document style, described in the manual, should be used for generating personal letters. For generating letters to be copied onto SRC letterhead, use the `srcletter` style. This style works for the old letterhead (the one with the Digital logo printed in blue). Mailing labels are formatted in two columns of five 2"×4-1/4" labels each, suitable for copying onto Avery brand, number 5352 address labels.

There are no features for making letters other than those described in the manual. However, suggestions will be accepted for such options as the ability to print ones net address as part of the letterhead.

## 2.2.5 The `showidx` Style Option

This style option, for use with the `report` or `book` document styles, causes index entries to be printed in the outer margin. It does not change the effect of `\makeindex`, which controls whether or not an `.idx` file is written. No attempt is made to avoid overprinting marginal notes. This option issues a `\flushbottom` declaration.

## 2.2.6 The `preview` Style Option

This option formats the output so it will appear as a continuous scroll when viewed with the *Proof* preview program. (See the *man* page for for a description of *Proof*.)

## 2.3 The `ps-slides` Style

This style causes `SLIT` to produce output that can be printed on the QMS color printer with `aptex` to produce color transparencies. When the output is printed on a black-and-white PostScript printer, colors are printed as different shades of gray.

The `ps-slides` style has the following differences from the ordinary `slides` style.

The `\colors` command is made a no-op. Colors that are not predefined must be defined with the `\newpscolor` command (see below).

The `\colorslides` and `\blackandwhite` commands are the same, except that `\colorslides` prints only slides and overlays, not notes.

The meaning of the `\invisible` declaration has been changed, as described below. The old `\visible` command is the same as the new `\white` command.

The default page style is `plain`. (The alignment marks are no longer of any use.)

The positioning of the body of the slide and the slide number have been changed to accomodate the QMS printer's small effective page size.

Except for these differences, a `SLI T -.1667emEX` input file prepared for the ordinary `slides` style *should* work with the `ps-slides` style. There are probably some `SLI T -.1667emEX` commands that will interact incorrectly with color-changing commands. These problems will be corrected when reported. The most likely bug is a color declaration kescaping from its scope, having a more global effect than it should. Such a problem can be worked around by adding redundant color-changing commands.

The new commands defined by the `ps-slides` style are listed below.

```
?newpscolor?cmd??red??green??blue?  
?renewpscolor?cmd??red??green??blue?
```

Define the command `cmd` to produce a new color with the indicated RGB values, where *red*, *green*, and *blue* are decimal numbers between 0 and 1 (inclusive). For example:

Black	0	0	0
White	1	1	1
Red	1	0	0
Light yellow	.5	.5	0

For `\newpscolor`, `cmd` must not already be defined; for `\renewpscolor`, it must already be defined (but not necessarily as a color). The following colors are predefined:

```
\black, \red, \green, \blue, \yellow, \magenta, \  
cyan, \white
```

For ordinary printers, which assume white paper, `\white` text is invisible. Unlike in ordinary `SLI T -.1667emEX`, color-changing commands can be used in math mode.

```
?invisible  
?visible
```

Ordinary declarations of visibility of the text. Invisible text is not printed. (TEX prints it in a font that whose letters are all print as spaces.) Visibility is independent of color, so you can have invisible red text, visible green text, and even visible white text. (Visible white text can be seen only when printed by a weird printer that prints on nonwhite paper.)

```
?norestore  
?restore
```



The `\norestore` declaration inhibits the proper scoping of color declarations, causing color declarations to act as if they were global. The `\restore` declaration causes the current and future colors to become the ones they would have been had there been no `\norestore` command. Both `\norestore` and `\restore` are global declarations. Example:

```
\black black \green green
\norestore \red red red
\restore
green
black
```

The `\norestore` declaration is handy for color commands inside a `tabbing` or `tabular` environment. However, there are some anomalies:

A `\restore` command inside a `tabbing` environment may not work exactly as it should. Try putting the command immediately after a `\=`, `\>`, or `\>`.

An `\fbox`, `\framebox`, or `\frame` command may do weird things in the scope of a `\norestore` if there's a color declaration in its argument.

A `\background` may behave strangely inside the scope of a `\norestore`.

?background?color??text?

Typesets *text* in an with a box behind it whose color is determined by the *color* color-changing command. The yellow box is the width of *text* plus a border of width `\bgborder` around it.

?hollowbackground  
?filledbackground

A `\hollowbackground` declaration causes any `\background` command in its scope to produce an outline, much like `\fbox`, instead of a filled box. This is useful for checking the slides with `Proof` or on a black-and-white printer. A `\filledbackground` declaration has the opposite effect.

?nogray

A declaration that makes every color other than `\white` be equivalent to `\black`, and issues a `\hollowbackground` declaration. Used to print the slides on a black-and-white printer so colors come out black instead of various shades of gray.

## 2.4 Where the Files Are

All L<sup>A</sup>T<sub>E</sub>X files mentioned in the manual, including the `sty` and `doc` files, are in the directory `/usr/local/lib/tex82`. Fonts are stored in two directories: the `tfm` files used by `TEX` are on `/usr/local/fonts/tfm`, and the pixel files used by `iptex` and other device drivers are in `/usr/local/fonts/pixel`.

## 2.5 Running `lablst.tex` and `idx.tex`

A list of labels and citations in an input file is printed by running L<sup>A</sup>T<sub>E</sub>X on the input file `lablst.tex`, which is done by typing

```
latex /usr/local/lib/tex82/lablst
```

L<sup>A</sup>T<sub>E</sub>X will then ask for the name of the input file, which should be typed without an extension, and for the name of the main document style (e.g., `article`), used by that file.

The index entries on an `idx` file are printed by running L<sup>A</sup>T<sub>E</sub>X on the file `idx.tex`, which is done by typing

```
latex /usr/local/lib/tex82/idx
```

L<sup>A</sup>T<sub>E</sub>X will ask for the name of the `idx` file, which is typed without an extension.

## 2.6 Differences from the Manual

All L<sup>A</sup>T<sub>E</sub>X features described in the manual are provided by the implementation at SRC.

## 2.7 Using B<sub>I</sub>B T<sub>E</sub>X

B<sub>I</sub>B T<sub>E</sub>X is a program for compiling a reference list for a document from a bibliographic database. It is run by typing

```
bibtex myfile
```

where `myfile.tex` is the name of your L<sup>A</sup>T<sub>E</sub>X input file. This reads the file `myfile.aux`, which was generated when you ran L<sup>A</sup>T<sub>E</sub>X on `myfile.tex`, and produces the file `myfile.bbl`. B<sub>I</sub>B T<sub>E</sub>X should be run from the directory containing `myfile.tex` (which should be the same directory from which L<sup>A</sup>T<sub>E</sub>X was run on that file).

If the `bib` file is not in the same directory as the L<sup>A</sup>T<sub>E</sub>X input file—for example, if you're using someone else's `bib` file—then you must include a path as part of the file name specified by the `\bibliography` command. A `~` cannot appear in the argument of a `\bibliography` command, so you should use a complete path name. For example, the L<sup>A</sup>T<sub>E</sub>X command

```
\bibliography{/udir/jones/bibfiles/gnus
```

specifies the file `gnus.bib` kept by Jones in his `/bibfiles` directory.

There is now no formal provision for sharing bibliographic database information, nor are there programs to assist in making your own `bib` files. Suggestions for forming one or more common `bib` files are welcome.

In addition to the bibliography styles described in the manual, there is a `ieeetr` style that formats entries in the style of the IEEE transactions.

In addition to the usual three-letter abbreviations for the months, the following abbreviations are defined by the bibliography styles:

```
0pt -.5 =2pt plus 1pt
acmcs      ACM Computing Surveys
acta       Acta Informatica
cacm       Communications of the ACM
```

ibmjrd	IBM Journal of Research and Development
ibmsj	IBM Systems Journal
ieeese	IEEE Transactions on Software Engineering
ieeetc	IEEE Transactions on Computers
ieeetcad	IEEE Transactions on Computer-Aided Design of Integrated Circuits
ipl	Information Processing Letters
jacm	Journal of the ACM
jcscs	Journal of Computer and System Sciences
scp	Science of Computer Programming
sicomp	SIAM Journal on Computing
tocs	ACM Transactions on Computer Systems
tods	ACM Transactions on Database Systems
tog	ACM Transactions on Graphics
toms	ACM Transactions on Mathematical Software
toois	ACM Transactions on Office Information Systems
toplas	ACM Transactions on Programming Languages and Systems
tcs	Theoretical Computer Science

## 2.8 Using `SLIT -1667emEX`

`SLIT -1667emEX` is a version of `LATEX` for making slides. To run `SLIT -1667emEX` with a root file `myroot.tex`, you type

```
slitex myroot
```

(You should be connected to the directory containing `myroot.tex`.) Refer to Section ? if you want slide files or `\input` files to be in a different directory from your root file.

Color slides can be made by copying the color layers produced by `SLIT -1667emEX` onto color transparencies using the Thermofax machine on the second floor. However, it is easier to make them with the QMS color printer on the 3rd floor. To use it, you must use the `ps-slides` document style described in Section ?. The output of `SLIT -1667emEX` is then printed by the command

```
aptex -Pcolor file-name.dvi
```

Before executing this command, check that the printer has transparencies in the paper tray and perform the appropriate protocol to synchronize with other users of the printer.

## 2.9 Using *MakeIndex*

The *MakeIndex* program helps in making an index. It is described in a separate document, available through the `printdoc` command.

## 2.10 Fonts

Almost all the symbols available on our fonts can be generated by ordinary L<sup>A</sup>T<sub>E</sub>X commands. However, there are type sizes not obtainable by L<sup>A</sup>T<sub>E</sub>X's size-changing commands with the ordinary document styles. Consult a local T<sub>E</sub>X expert to find the T<sub>E</sub>X name for such a font.

Tables ? and ? allow you to determine if the font for a type style at a particular size is preloaded, loaded on demand, or unavailable.

size	default (10pt)	11pt option	12pt option
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Table 1: Type sizes for L<sup>A</sup>T<sub>E</sub>X size-changing commands.

	?it	?bf	?sl	?sf	?sc	?tt
5pt	D	D	X	X	X	X
6pt	X	D	X	X	X	X
7pt	P	D	X	X	X	X
8pt	P	D	D	D	D	D
9pt	P	P	D	D	D	P
10pt	P	P	P	P	D	P
11pt	P	P	P	P	D	P
12pt	P	P	P	P	D	P
14pt	D	P	D	D	D	D
17pt	D	P	D	D	D	D
20pt	D	D	D	D	D	D
25pt	X	D	X	X	X	X

Table 2: Font classes: P = preloaded, D = loaded on demand, X = unavailable.

Table ? tells you what size of type is used for each L<sup>A</sup>T<sub>E</sub>X type-size command in the various document-style options. For example, with the 12pt option, the `\large` declaration causes L<sup>A</sup>T<sub>E</sub>X to use 14pt type. Table ? tells, for every type size, to which class of fonts each type style belongs. For example, in 14pt type, `\bf` uses a preloaded font and the other five type-style commands use load-on-demand fonts. Roman (`\rm`) and math italic (`\mit`) fonts are all preloaded; the `\em` declaration uses either italic (`\it`) or roman.

## 2.11 Using Times Roman Fonts

The fonts normally used by L<sup>A</sup>T<sub>E</sub>X are from the Computer Modern Roman font family designed by Donald Knuth. Some people prefer the Times Roman family of fonts, which are available on the Apple LaserWriter printers. They are obtained with the `ps` document-style option. Currently, this option does not work with the `12pt` style option. Also, fonts from the Times Roman family are available only for the default `\rm` style and for the `\it`, `\bf`, and `\sf` styles. Other styles use the ordinary CMR fonts.

TEX can use only the CMR fonts in math mode, so `$_\rm foo=3$` produces `kfoo` in CMR font, which is probably not what you want. Instead, use `$_\mbox{\rm foo=3}$`, which typesets `kfoo` in Times Roman because TEX is not in math mode inside the `\mbox`. No one will notice that ordinary math symbols like `*` or `x` are typeset in a CMR font rather than a Times Roman font.

A `dvi` file produced with the `ps` option can be printed only on the Apple LaserWriter using the `aptex` program. Type the Ultrix command `man aptex` to find out how to run this program. You can print any `dvi` file with `aptex`, but it is slower than `iptex` and doesn't do a very good job printing the standard CMR fonts. So, use `aptex` to print only `dvi` files created with the `ps` option.

The `ps` option causes L<sup>A</sup>T<sub>E</sub>X to use more fonts than usual, so there's a good chance that you'll run TEX out of font space if you use lots of different fonts.

## 2.12 Special Versions

No foreign-language or other special versions of L<sup>A</sup>T<sub>E</sub>X are currently available at SRC.

## 3 Bugs

There are a few known bugs in L<sup>A</sup>T<sub>E</sub>X that occur very seldom and cause the user little trouble, but would be very difficult to fix. Moreover, given the nature of complex systems, it is not unlikely that the corrections would lead to even worse problems. Therefore, these bugs will probably not be fixed.

The bugs and ways to get around them are listed below. Do not worry about any of them until you are preparing the final draft, since changes to the text are very likely to cause the problem to disappear.

In rare instances, a figure or table will be printed on the page preceding the text where the `figure` or `table` environment appears. This can be fixed by either moving the environment further towards the end of the document.

A marginal note at the top of a page may appear in the wrong margin. This can be fixed by inserting a redundant `\pagebreak` command to force a page break exactly where L<sup>A</sup>T<sub>E</sub>X started the new page anyway.

A footnote can be broken across two pages when it should fit on a single page. This happens when there is one or more figures or tables on the page. The problem is corrected by moving, towards the end of the file, the last `figure` or `table` environment that produces a figure or table on the page where the footnote starts.

## 4 Errata and Additions to the Manual

These are all the errors and omissions to the manual, *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System* reported by 6 June 1989.

### 4.1 First Four Printings

The following errors were corrected in the Fourth Printing. To see if your copy is one of the first three printings, turn to the copyright page the one with the Library of Congress cataloging information, on the back of the title page. If the fifth line on that page begins kthat, as of August 27, 1985, then your copy is from the first three printings.

**page xiii**

Add Mike Urban to the list of people thanked in the third paragraph.

**page 2**

In the first paragraph of Section 1.1, replace the three instances of `sample.tex` by `small.tex`.

**page 15, line 18**

Replace `kthayl` by `kthatl`.

**page 24, line 13**

Add a period to the end of the line.

**page 31, line 7**

Delete one `kwilll`.

**page 39, line 2**

Change `ksixl` to `kfivel`.

**page 39, line 17**

Remove the extra `kitl`.

**page 40, line ?8**

Change `kfivel` to `ksixl`.

**Section 3.3.5, page 49**

The name of the environment is `eqnarray`. There are two instances on this page of the incorrect name `keqnarryl` that should be changed.

**page 52, last line**

Replace `kinstead of l` by `kinstead of l`.

**pages 55m58, 124, and 173m174**

Commands that define or redefine a command or environment, such as the `\newcommand` and `\renewenvironment` commands, should not be nested within one another. Doing so may result in the following TEX error:

```
! Illegal parameter number in definition of ... .
```

**page 56, line ?3**

Change `k\bf x l` to `kxl`.

**page 60, line ?7**

Replace `kone of these environmentsl` by `ka figure or table environmentl`.

**page 74, line 20**

Change `\begindocumentstyle` to `\begindocument`.

**page 75, line 15**

Change `\thebibliography` to `\bibliography`.

**page 79, line 16**

Change `kdiplayingl` to `kdisplayingl`.

**page 88, line 27**

Change `kentire the paragraphlto kthe entire paragraphl`.

**page 89, line 16**

Change `kpage-breakingl` to `kline-breakingl`.

**page 95, line 5**

Replace `k.01inl` by `k1.01inl`.

**page 96**

In the penultimate paragraph of Section 5.4.2, replace

The `\vfill` command is an abbreviation for `\vspace\fill`.

with the following:

The `\vfill` command is equivalent to a blank line followed by `\vspace\fill`.

**page 98, line 6**

Remove an `kil` from `kdirectionl`.

**page 102**

In Figure 5.1, replace the two occurrences of `k-1.8l` by `k?1.8l`.

**page 104, line ?3**

Correct the spelling of `kpositioningl`.

**page 113**

Change the label on the arrow above Item 2 from `\itemsep` to `\itemsep + \parsep`.

**page 116, line 20**

Change `kfourteen-point Plus Romanl` to `ktwelve-point Plus Romanl`.

**page 118, line ?10**

Change this line to:

```
... (myfile.tex [1] [2] [3] (part1.tex [4] [5]) (part2.tex [6] [7]
```

**page 129**

The following L<sup>A</sup>T<sub>E</sub>X warning messages should be listed:

No ... typeface in this size, using ...

A type style declaration specified a type style and size combination that is not available, so L<sup>A</sup>T<sub>E</sub>X is substituting another one.

... in math mode.  
The indicated command is not permitted in math mode but was used there.  
Remember that `\boldmath`, `\unboldmath`, and size-changing commands may not be used in math mode.

**pages 141m142**

Change `kvon Beethovenl` to `kvan Beethovenl`.

**page 142**

Add the following near the bottom of the page, just above the **Titles** heading.

If an `author` or `editor` field has more names than you want to type, just end the list of names with `and others`; the standard styles convert this to the conventional *et al.*

**page 147, line 6**

Change `kScribe-compatiblel` to `kScribe-compatiblel`.

**page 152, line ?6**

Add `\newtheorem` and `\hyphenation` to the list of global declarations.

**page 157, line ?10**

The first paragraph of Section C.3.1 should note that the \*-form of a sectioning command does not affect the running head.

**page 158, line ?12**

Change `kchaptersl` to `kappendix chaptersl`.

**page 160, top line**

Change `k(Section 5.6)l` to `k(Section 5.3)l`.

**page 160, line ?9**

Change the description of the `openbib` style option to: `kCauses the bibliography (Section 4.3) to be formatted in open style. (See van Leunen [7].)l`

**page 163, line 19**

Change `kSection 5.1.2l` to `kSection 5.1.1l`.

**page 168, line ?5**

Replace `kprinted as \ l` by `kprinted as l`.

**page 169, line ?5**

Change `kfirst and third rowsl` to `kfirst and third columnsl`.

**page 169, line ?3**

Add a period to the end of the sentence.

**page 170**

Add the following paragraph after line 5:

An `overfull \hbox` warning occurs if a formula extends beyond the prevailing margins. However, if the formula does lie within the margins, no warning is generated even if it extends far enough to overprint the equation number.



**page 174, line ?1**

Add the following sentence: kIt is a global declaration.l

**page 176**

Change the four lines immediately following the heading for Section C.8.1 to:

```

\beginfigure[loc]                body          \endfigure
\beginfigure* [loc]              body          \endfigure*
\beginable[loc]                  body          \endtable
\beginable* [loc]               body          \endtable*

```

page 185, line ?5

Replace ksuppresses commandl by kcommand suppressesl.

**page 187, lines 15m17**

The sentence kIt also writes *bib\_files*?l is redundant and can be eliminated.

**page 188, lines 1, 2, 4, 8**

Change *kkey\_lstl* to *kkey\_listl*.

**page 191, line 13**

The word kparagraphl misspelled.

**page 191, line 14**

Remove the space between \begin and sloppypar. (This is for consistency only; LATEX ignores the space.)

**page 199**

Add the following sentence to the last paragraph on the page:

Words typeset in typewriter style or in two different styles are not hyphenated except where permitted by \- commands.

This is a change to LATEX made on 18 December 1985.) Also, add the following index entry citations to this page: k\-, khyphenation, ksuppressedl, and ktypewriter type style, no hyphenation inl.

**page 217**

Add the subentry kopenbib, 160l to the index entry kdocument-style optionl.

**page 223, column 2, line 13**

Change kmadl to kmadel.

**page 225**

Add page 47 to the index entry for kLeunen, Mary-Claire vanl.

**page 228**

In index entry for \multicolumn, change k194l to k184l.

**Tear-Out Command Sheet**

In the first column, sixth line after kSentences and Paragraphsl heading, replace k& &l by k& \&l.

## 4.2 First Five Printings

### page xiii, line ?7

The hyphenation of `kLeunenl` is obviously incorrect.

### page 7, line 33

Add the following sentence after `k[6].l`

( Many of the details of L<sup>A</sup>T<sub>E</sub>X, including command names, were also taken from *Scribe*.)

### page 12, line ?7

Replace `keveryl` with `keverl`.

### page 53, line 16

Change `kmath italic` is the default in math model to `kmath italic` is almost identical to the default type style of math model.

### page 71, last line

Change `kcomandl` to `kcommandl`.

### page 84

Add the following paragraph just after line 2.

Some commands, such as `\chapter`, change the style of the current page. You can change it yourself with the `\thispagestyle` command.

Also, add the appropriate index entries for `\thispagestyle` and `\chapter`.

### page 88, line 29

Change `kitsl` to `kitjsl`.

### page 91, line 5

It should be noted that in the `book` document style, `\chapter` uses `\cleardoublepage`.

### page 111, line ?4

Remove the extra `t` from `kwithtoutl`.

### page 122, line 17

Change the sentence that begins `kThe most probable cause isl` to the following:

The most probable cause is a list-making environment with no `\item` command.

### Page 126, line 12ff

An `\protected` command even a robust one in the argument of one of these commands can cause TEX to run out of this kind of space. It's a good idea to `\protect` all commands in these arguments.

### Page 153, top of page

It should be noted that if the invisible `kwordl` occurs at the end of a paragraph, not attached to a real word, then it could appear on a line by itself, producing a blank line in the output.

**Page 160, line ?7**

Replace `kequationsl` with `kequationl`.

**Page 162**

In the description of `\pagenumbering`, it should be mentioned that the command sets the value of the `page` counter to 1.

**Page 161, line ?9**

Replace `khead informationl` by `khead information` (but not page number)l

**Page 162, line 6**

Replace `kheadingl` by `kheadingsl`.

**Page 177, line 1**

It should be noted that, in a two-column page style, a single-column figure can come before an earlier double-column figure, and vice-versa.

**Page 177, line ?1**

Replace `kpagel` with `ktext pagel`.

**Page 178, lines 1 and 8**

Replace `kpagel` with `ktext pagel`.

**page 183**

The description of `\extracolsep` should mention that no extra space is put to the left of the first column.

**page 191**

Add to the description of `\pagebreak` the information that it has an effect only when used in paragraph mode, not inside a `parbox`.

**page 191, line ?2**

Add a `kti` to `kexplicilyl`.

**page 213**

Change the page reference for `k\bibitem`, moving argument `ofl` from 153 to 152.

**page 211**

Change index entry for `\AE` to page 40.

**page 213**

Change the first page reference for `kbb1` (bibliography) filel from 74 to 75.

**page 217**

Add an index entry `kdiacritical marks`, see `accentsl`.

**page 219, index entry for kfiguresl**

Remove the `kandl` from the subentry for `koutput` by `\enddocumentl`.

**page 219, column 1, line 3**

Replace `kanomolousl` by `kanomalousl`.

**page 223, left column, line 2**

Remove the page number k79l from the first subentry under `\index`.

**page 228**

Change the page reference for `k\multicolumnl` from 194 to 184.

**page 228**

Remove the extra `k,l` from the index entries for `\o` and `\O`.