

L^AT_EX Update

Leslie Lamport

Version 2.09 —

This corrects all known significant inaccuracies in the December 13, 1983 Version of the L^AT_EX manual, including both typographic errors and changes to L^AT_EX.

This document was extensively revised on November 8, 1984. After this revision, all subsequent material will be added only at the end.

References to pages in the manual refer to the December 13, 1984 printing. Subsequent printings have slightly different pagination. Line $-n$ denotes the line from the bottom of the page.

Errata

There are many typos and omissions in the preliminary manual. Most of them are obvious and you can correct them yourself. The following are the only ones that may cause trouble.

page 10, line -9

Typing `<` and `>` do not produce the expected results. These characters should be typed only in math mode.

page 22, line 9

“right to left” should read “left to right”.

page 41

The manual should mention that a list environment must contain at least one `\item` command.

page 48, second paragraph

Ignore this paragraph. There are cases in which omitting the braces can cause trouble.

page 77, lines 18 and 22

Delete the `#`; the optional argument should just be 1.

page 79, line -6

Replace `\textwidth` by `\textheight`.

page 90

If a `minipage` environment contains nothing but one or more `tabbing` environments, then the width of the resulting box is the minimum of: (i) the argument of the `minipage` environment and (ii) the width of the widest line in the `tabbing` environment(s). This may be useful for making a `\parbox` whose width is determined by its contents.

page 103ff

An `array` and `tabular` environment leaves a small space (half the normal intercolumn space) at the left and right of the array. A `|` or `@`-expression at the beginning (end) of the argument suppresses the space at the left (right). This space usually does no harm, but can be eliminated by putting `@{ }` at the beginning and end of the argument.

page 118, line -12

There should be three more spaces before “Special characters:”.

Page 128

Add a reminder about typing `\twocolumn[{ ... }]` if the argument contains a `]`.

Page 138

Replace `\bibstyle` by `\bibliographystyle`.

Changes Through Version 2.05

pages 46ff

A new `eqnarray` environment has been added. Writing

```
\begin{eqnarray} ... \end{eqnarray}
```

is pretty much the same as writing

```
\[ \begin{array}{lcl} ... \end{array} \]
```

except that:

- An equation number is put on every line, unless there is a `\nonumber` command on that line.
- Page breaking can occur between lines, so the `*`-form of the `\\` command is useful.
- There is an extra bit of space added between the lines. The amount of space is equal to the length `\jot`, which is a parameter you can set.

There’s also a `eqnarray*` environment which is the same as `eqnarray` except it doesn’t put in any equation numbers.

The `equation`, `eqnarray` and `eqnarray*` environments are called *equation* environments. You can put them inside `parboxes`, but you should not put an equation environment inside a `parbox` that’s inside an equation environment.

Section 2.3.2

The following math symbols have been added. (Like all math symbols, they can be used only in math mode.)

\mathbb{O}	<code>\mho</code>	\sqsubset	<code>\sqsubset</code>
\mathbb{X}	<code>\Join</code>	\sqsupset	<code>\sqsupset</code>

page 48

Upper-case Greek letters come in the same type styles as ordinary Roman letters. In math mode you can type `{\bf \Pi}` to produce $\mathbf{\Pi}$, `{\tt \Pi}` to produce \Pi , and so on.

pages 55, 72ff, and 142

You can now put “dangerous” commands in an @-expression, or in the argument of a sectioning or `\caption` command, by preceding the command with `\protect`. For example, you can write

```
\begin{array}{l@{\protect\makebox[.2in]{=}}
l} ...
```

or

```
\section{Figure \protect\ref{foo} Re-examined.}
```

The `\writecommand` command has been eliminated, and `\protect\foo` does what `\writecommand{\foo}` used to.

The `\protect` command can be used in the argument of a `\typeout` or `\typein` command, so

```
\typeout{Command \protect\foo?}
```

causes T_EX to type

```
Command \foo?
```

on the terminal. Up until version 2.07, you can also use `\protect` in the argument of an `\index` command to write a command onto the `.IDX` file. (See the change to `\index` in version 2.07.)

An extra `\protect` command does no harm, so you can use one when you’re not sure if it’s necessary.

page 60

The declaration `\boldmath` has been added. It changes the math italic typeface and the math symbols to boldface. (This makes lowercase Greek letters bold, but not

uppercase ones.) The declaration `\unboldmath` undoes its effect. These declarations are like size-changing declarations, and they should *not* be used in math mode. The `\boldface` declaration has some anomalies in terms of what it makes bold and what it doesn't. Also, it does not embolden subscripts and superscripts.

page 99

A `tabbing` environment should not appear inside another `tabbing` environment. (This would be possible only if the inner environment were inside a `parbox`.)

pages 117-118

This change has been made obsolete in version 2.07 by the `\newenvironment` command.

The `\newlist` command now has an optional argument that may come right before the last argument. It specifies the number of arguments that the new list environment has. (The default is, as before, no arguments.) These arguments can be mentioned in the last mandatory argument just as in the `\newcommand` command. For example, you can write:

```
\newlist{labelquote}{}{}[1]{\item {\bf #1:} }
```

to define a `quote`-like environment which typesets its argument in boldface in front of the `quote`'d material.

page 125

In $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ version 2.01, the `\mainbaselineskip` command has been eliminated. In its place is the command `\baselinestretch` that works as follows. Every command that changes the type size sets the value of `\baselineskip` to `\baselinestretch` times its “ordinary” value for that type size. For example, the ordinary `\baselineskip` for a ten-point font is 12pt. If `\baselinestretch` equals 1.5 and the `\normalsize` type size is ten-point, then the `\normalsize` command sets `\baselineskip` to $1.5 \times 12\text{pt}$, or 18pt.

You can change `\baselinestretch` at any time with a `\renewcommand` declaration, but remember that this has no immediate effect on the value of `\baselineskip`; it affects only later type-size-changing commands. The `\begin{document}` command executes a `\normalsize` command to initially set the type size, so changing `\baselinestretch` before the `\begin{document}` command has the effect of “magnifying” `\baselineskip` throughout the document.

page 138

The `thebibliography` environment has been changed slightly. The `\bibitem` command automatically puts brackets or whatever around the labels in a bibliography, so the example on line -7 should read

```
\bibitem[Knuth83]{knuth-tex82} Knuth, Donald ...
```

The document style determines if the label is to be printed as “[Knuth 83]” or “Knuth 83.” or whatever. The argument of the environment itself should be the same as the optional argument of a `\bibitem` command to generate the longest label.

page 146

The error message

```
! Something's wrong--perhaps a missing \item.
```

has been added. The most likely cause is forgetting an `\item` command in a `list` environment. It is also caused by forgetting the argument of a `thebibliography` environment.

page 146

A new \TeX error is now possible:

```
! Font ... not loaded: Not enough room left.
```

This means that you are using too many different typefaces, where using the `\it` type style in `\small`, `\normalsize` and `\large` counts as three different typefaces. You should probably eliminate some of those typefaces, since they’re just going to confuse the reader anyway. However, you might solve your problem by \LaTeX ing your document in parts.

page 148

The error message

```
! This is a LaTeX bug.
```

has been changed to

```
! This may be a LaTeX bug.
```

It is possible, though difficult, to get it with bad input.

pages 148,152

The “Typeface not available” error has been eliminated. If a typeface isn’t available, L^AT_EX uses a different font and types out the warning:

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

Page 160 (addendum)

Text in which there is no color declaration in effect appears on all color layers. For example, if you make no color declarations anywhere in your slide file, then all color layers will be identical to the black and white versions. Note that color declarations are undefined in the root file.

Page 161 (addendum)

The warning about horizontal lines appearing in color layers where they should not can be eliminated. Colors should now always be handled properly.

Changes Made in Version 2.06

pages 8, 119, 120, etc.:

The `\pagelayout` command has been eliminated; the parameters that used to be specified by the page layout are now specified by `\documentstyle`. The `\documentstyle` command has been changed. It now has the following form:

```
\documentstyle[ options]{style}
```

where *style* is the main style and *options* is an optional list of options, separated by commas. The main styles and available options are:

<u>Main Style</u>	<u>Options</u>
report	11pt, 12pt, twoside, twocolumn, draft, fleqn
article	11pt, 12pt, twoside, twocolumn, draft, fleqn
letter	11pt, 12pt, fleqn
slides	—

¹The `fleqn` option makes all displayed equations flushleft, a distance of `\mathindent` from the prevailing left margin.

L^AT_EX only.

page 10

You can forget about the `\mbox` nonsense for producing a sentence-ending period. You simply type " (that's a double-quote) *before* the period, so the example should be typed as follows:

```
Euclid et al.\ proved I + I = II". Meanwhile ...
```

You can use the " in this way with other punctuation marks that ordinarily have extra space added after them, including ? and :. You should also do the same thing before a closing parenthesis, as in:

```
... I + I = II"?) Meanwhile ...
```

Note that you should not have any other "s in your text (except inside a verbatim environment or a `\verb` command).

In version 2.07, you type \@ instead of ".

pages 11, 72ff

Two new levels of sectioning below `\subsubsection` have been added: they are `\paragraph` and `\subparagraph`, together with their *-forms. Two new counters have also been added to control which levels of sectioning are to be numbered and which are to be listed in the table of contents. Setting the counter `secnumdepth` to k ($k \geq 1$) means that section levels down through level k (inclusive) are numbered, and setting the counter `tocdepth` to k means that sections at levels down through k are to appear in the table of contents. In both the `report` and `article` document style, `\section` is the level-1 command, `\subsection` the level-2 command, and so on. (The restrictions on what may appear in the argument of a sectioning command given on page 72, as modified by the introduction of the `\protect` command, apply even if `tocdepth` specifies that no table-of-contents entry is to be produced.)

pages 19, 32

The declaration `\em` chooses a font style for *emphasis*—either *italic* or *roman*, depending upon where it is used. Thus, typing

```
Remember: {\em this is {\em very} important.}
```

produces

Remember: *this is very important.*

It is best to use `\em` instead of `\it` for two reasons: (i) you might later decide to italicize a paragraph and forget that to convert `\it` commands to `\rm` commands,

and (ii) there are environments like the `theorem` environment which some document styles might set in roman type and others in italic.

pages 41,42

The `description` environment no longer takes an argument. It also formats things differently—the label is now italicized and flushed right. You aren't supposed to worry about the formatting— $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ is supposed to take care of that for you. However, if you must fiddle with it, look up the definition of the `description` environment in the `.DOC` versions of the style files.

pages 54, 55, 72, 84

The `\s` command has been eliminated. You can now use an `\hspace` command wherever you had to use the `\s` command.

page 75

A new feature for making a title has been added. The command `\maketitle`, which makes a title for your document which goes on a separate page for the `report` style and goes at the top of a page for the `article` style. (The `titlepage` option for the `article` style causes the title to be on a separate page.) The components of the title are declared with the following commands:

- `\title` declares the title. You may use `\\` to generate a new line.
- `\author` declares the author. Use `\\` to start new lines—for example, between the author's name and his address.
- `\date` declares the date. If omitted, today's date is used.

A `\thanks` command may be used in the title, author, or date. It acts pretty much like a `\footnote` command, and is usually used to attach an acknowledgement to an author's name. These declarations may come anywhere before the `\maketitle` command, as in the following example.

```
\title{This is Your Life}
\author{John Jones\thanks{Work supported by ...}
and Fred
Smith\thanks{Work supported by ...} \\
Institute for Inhuman Development}
...
\begin{document}
\maketitle
```

You can use at most one `\maketitle` command in your document.

See also the `\and` command for multiple authors and the abstract environment, defined in Version 2.07.

page 82

The `\newnumbering` command has been eliminated. I will illustrate how counter numbering works using the `section` counter as an example. The section number, as it appears in the table of contents or in the section heading, is produced by the command `\thesection`. You can change how sections are numbered by redefining this command. For example, assuming that chapters are numbered I, II, ..., to get sections numbered II.A, II.B, II.C, ... you can type the following

```
\renewcommand{\thesection}{\thechapter.  
\Alph{section}}
```

page 88

The naming of “bins” for saving boxes with the `\savebox` command has been changed to conform to the same convention as the naming of lengths. There is a `\newsavebox` declaration that defines a bin. Instead of writing a bin number, the `\savebox` and `\usebox` commands take a bin name. For example, you write

```
\newsavebox{\foo} ...  
\savebox{\foo}[lin]{gnus} ...  
\usebox{\foo}
```

When you leave the scope of a `\newsavebox` declaration, the contents of the box vanishes. This is a handy way to reclaim the space taken up by the contents of a bin. However, if you don’t put the `\newsavebox` command “downstairs”, but define it at “ground level”, then remember to “empty” the bin when you don’t need its contents any more by a command like

```
\savebox{\foo}{}
```

The old style `\savebox` and `\usebox`, with bin numbers, will still work as before, so you don’t need to change your current files. But you should use the new conventions in the future.

page 90

Eliminate the second warning—the one about beginning or ending a `minipage` environment with a paragraph-making environment. You can now put other environments inside a `minipage` with no problems.

page 115ff

The `list` environment has been changed in the following ways.

1. The `\topsep` parameter has been eliminated and a new (rubber) length `\partopsep` added. The space inserted above and below the environment is determined as follows:
 - If it begins a new paragraph, then `\topsep+\partopsep`.
 - If not, then `\topsep`.
2. The default settings of all the list parameters are now determined by the document style, and may vary with the nesting level of the list—e.g., being different for a list inside a list than for a top-level list. By convention, `\leftmargini`, ..., `\leftmarginvi` still determine the default value for `\leftmargin`.

page 120

There are now two versions of every document-style file. The T_EX “code” for the `report` document style still resides on the file `report.sty`, but there is also a file `report.doc` that has essentially the same code completely documented. (The file `report.sty` is obtained from `report.doc` by eliminating comments and extra spaces, and by adding some optimizations that make the T_EX code less readable.) The `.doc` files will make it easier for you to figure out how to customize your own document style.

page 120

A new command `\raggedbottom` has been added. It causes pages to assume their natural height instead of stretching them to all be the same height. It can be used anywhere, but it makes little sense to put it anywhere except before the `\begin{document}` command.

See the further amplification of this feature in Version 2.07.

page 151

Add the following error messages:

```
! \textfont ... is undefined (character ...).  
! \scriptfont ... is undefined (character ...).  
! \scriptscriptfont ... is undefined (character ...).
```

This is actually a L^AT_EX bug which has not been corrected because it happens very infrequently, and eliminating it would require adding some unpleasant restrictions to L^AT_EX. It occurs when using certain uncommon type-style declarations, like `\sc`, in math mode.

[The following has been made obsolete by the `\load` command introduced in Version 2.08.]

To get around the problem, you need simply add a ground-level (not inside any environment) declaration of the offending type style before the formula—preferably at ground level. Thus, you can put the commands `\sc \rm` somewhere before a math environment that uses the `\sc` command. The “dummy” `\sc` declaration should occur at the same type size as the one in the math environment, so a problem caused by

```
{\large $ ... {\sc foo} ... $ }
```

can be corrected by adding `\large \sc \normalsize`.

Changes in Version 2.06a

nowhere in particular

The command `\TeX` produces the T_EX logo, and `\LaTeX` produces the L^AT_EX logo.

Changes in Version 2.07

page 10

As noted above, the command `"` introduced in version 2.06 has been renamed `\@`. Thus, you type

```
Euclid et al.\ proved I + I = II\@.  
Meanwhile ...
```

to get the proper end-of-sentence space after the period.

page 36

The command `\pounds` produces a £ (pounds sterling) symbol.

page 52, Section 2.3.3 and pages 9-10

The `\,` command will work outside math mode, where it also produces a “thin” space. It is useful for quotes inside quotes, typing ```\, `foo'\, ''` to get “‘foo’”.

page 59

The `\ldots` command now works in any mode, not just in math mode.

page 62ff

You can now cross-reference footnotes by putting a `\label` command in with the text of the footnote, as in

```
this is it.\footnote{\label{its-it}It really is.}
```

page 75

A new feature has been added to the title-making macros. Multiple authors in an `\author` command should be separated by an `\and` command, as in

```
\author{T. Jones \\ SRI International \and J. Smith \\ Stanford}
```

Note that multiple-line “authors” are made with the `\\` command, as expected.

There is also an `abstract` environment for producing the abstract of the paper. It should normally follow the `\maketitle` command. You can still roll your own abstract if you want, but the `abstract` environment makes life easier when changing document styles.

pages 81ff, 115ff

The method of implementing label numbers in lists has changed, making it easier to define your own list environments. The `list` and `enumerate` counters have been eliminated, and two new commands have been added. First, there’s a `\newcounter` command, where `\newcounter{foo}` defines a new counter named `foo`. There’s an optional second argument, where writing

```
\newcounter{foo}[subsection]
```

causes the counter `foo` to be reset to zero whenever the `subsection` counter is incremented.

The second relevant command has the form `\usecounter{foo}`, where `foo` is any previously defined counter. This command is used in the second argument of

the `list` environment to allow counter `foo` to be used to number the list items. More precisely, it causes `foo` to be initialized to zero and incremented by every `\item` command that has no optional argument. A `\label` command within an item will refer to the value of counter `foo` for that item.

The `enumerate` environment is implemented with the counters `enumi` for an outermost environment, `enumii` for the first nested inner `enumerate`, ... and `enumiv` for a fourth-level enumeration.

If you've played around with the document styles to modify any of the list environments, you'll have to look at the new `.doc` files to see how the inner workings have been changed. (It's now easier to modify the way enumerations are numbered and referenced.)

page 82

In addition to `\arabic...` `\Alph`, there's a new `\fnsymbol` command for printing counter values. It produces the the following sequence of nine symbols:

* † ‡ "278 "27B — ** †† ‡‡

that can be used for numbering footnotes. (The double symbols look better on footnotes than they do here.) Warning: the symbols it produces can be used only in math mode, so if you want to cross-reference something that's numbered in this way, you have to put the `\ref` command in math mode.

page 102, line -5

You will be able to type `\a``, `\a'`, and `\a=` to produce these accents inside a tabbing environment.

page 104

A new command has been added for drawing horizontal lines across only some of the columns of an `array` or `tabular` environment. The command `\cline{i-j}` draws a line across columns *i* through *j* inclusive. It is used just like `\hline`, except multiple `\cline` commands can be used to draw lines across distinct columns. For example

```
... \ \ \cline{1-3} \cline{5-5}
```

Draws horizontal lines across columns 1-3 and column 5.

pages 117-8

The `\newlist` command has been eliminated. In its place are two commands `\newenvironment` and `\renewenvironment` for defining arbitrary environments. To understand how to use these commands, you must first realize that

```
\begin{foo} ... \end{foo}
```

is essentially translated to

```
{\foo ... \endfoo}
```

The command

```
\newenvironment{foo}[3]{def1}{def2}
```

acts the way you'd expect

```
\newcommand{\foo}[3]{def1}\newcommand{\endfoo}
{def2}
```

to, except that `\newcommand` won't allow you to define a command name that begins with `\end`. (Of course, the “[3]” represents an optional argument.) The effect of `\renewenvironment` is similar.

The following paragraph is superseded by a change made in version 2.08.

You will often want the `\begin{foo}` to execute the `\begin` of some other environment and the `\end{foo}` to execute its `\end`. This doesn't work. The other environment must be called without using the `\begin{...}` and `\end{...}` commands, as shown below:

```
WRONG: \newenvironment{foo}{\begin{list}...}
{\end{list}}
RIGHT: \newenvironment{foo}{\list...}{\endlist}
```

pages 119 ff

A new document substyle `leqno` now exists. It causes equation numbers to appear on the left instead of the right. It works with all styles and substyles, including `fleqn`.

page 119

There is now a document style for producing camera-ready output for the ACM proceedings format—the large, two-column pages. The `acm` substyle of the `article` style produces output formatted for large (10×14) paper; the size known to the Imagen printer as B4. You should use the `12pt` substyle, so your file should begin with

```
\documentstyle[12pt,acm]{article}
```

Use the `\maketitle` command and the `abstract` environment to get the title and abstract in the right places. You can leave a space at the bottom of the first column for the copyright notice with the `\copyrightspace` command. It normally follows

right after the `\maketitle`. It works by producing a blank footnote, so it must follow any commands that produce footnotes in the first column. The `\head` command will cause its argument to be printed as an identification at the bottom of the page.

page 120

A new command `\flushbottom` has been added. It is the inverse of the `\raggedbottom` command introduced in version 2.06. It has been added because `\raggedbottom` is now the default in the `article` and `report` styles unless the `twoside` or `twocolumn` option is specified.

page 126

A new parameter `\arraystretch` has been added to control the interline spacing in the `tabular` and `array` environments. The spacing between rows will be `\arraystretch` times the current value of `\baselineskip`.

Note: The inter-row spacing in these environments is produced by putting a strut on each line rather than by $\text{T}_{\text{E}}\text{X}$'s normal baseline spacing mechanism. The `\[...]` command (with a positive argument) works by increasing the depth of this strut. This can fail to add the expected amount of extra space if there is something on the line that extends further down than the strut.

page 134

The argument of the `\index` command may now contain any character except a curly brace (`{` or `}`). That argument will be written onto the `.idx` file verbatim. The same applies to the `\glossary` command.

page 143 ff

You may get an obscure error after $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ has finished processing your input file, when the message file indicates that it is reading the `.AUX` file. Such an error is probably caused by a command in the argument of a sectioning or `\caption` command that needs to be `\protect`'ed.

page 146

The error message

```
! Can be used only in preamble.
```

has been added. It is caused by putting in the body of the text a command that belongs in the preamble (before the `\begin{document}` command).

page 146

The word `preamble` has been replaced by “array arg” in the following three error messages:

```
! Illegal character in preamble.  
! Missing p-arg in preamble.  
! Missing @-exp in preamble.
```

This was done only in later releases of Version 2.07.

Changes in Version 2.08

page 11, 71ff

The `\part` sectioning command has been added. It is the highest level of sectional unit and is optional. A `\part` command does not change the numbering of chapters or sections, so the first chapter of Part II could be Chapter 12. There is a `\part*` command that works like the `*`-form of the other sectioning commands.

page 28ff

A font representing an unusual type-size/style combination may not work right when used in math mode, either printing the wrong size characters or not printing any characters and generating one of the following error messages:

```
! \textfont          ... is undefined  
(character ...).  
! \scriptfont       ... is undefined  
(character ...).  
! \scriptscriptfont ... is undefined  
(character ...).
```

To solve this problem, use a command of the form

```
\load{size}{style}
```

where *size* is the size-changing command and *style* is the type-style command that together specify the desired font. The `\load` command should come before the first use of the font in math mode, and should not be inside braces or an environment.

page 32ff

The `\newfont` command `\newfont{\foo}{fontnm}` defines `\foo` to be a declaration that chooses the font named *fontnm*. You can't use `\foo` in math mode.

The command `\symbol{char}` produces the symbol with character code *char* in the currently selected font. Octal character codes are prefaced by `'` and hexadecimal codes by `"`, so you can type either `\symbol{26}`, or `\symbol{'32}`, or `\symbol{"1A}`, since .

pages 69,148

The `figure*` and `table*` environments are allowed in single-column format, being equivalent there to ordinary `figure` and `table` environments, so they produce no error message. When writing a paper, even if you're using a single-column format, it's a good idea to decide whether a figure or table should be single- or double-column if it were in two-column format. That way, the right thing happens if you ever decide to switch to double columns.

page 78ff

The length `\fill` has been added. It is the “rubber” length added by an `\hfill` command, so `\hspace{\fill}` is equivalent to `\hfill`. The command `\stretch{1.3}` produces a rubber space of length 1.3 times that produced by `\fill`. It can be used only as the argument of `\hspace` or `\vspace` (or their *-forms), which now can produce rubber spaces.

page 81ff

The command `\value{foo}` produces the value of the counter `foo`. It can be used where L^AT_EX expects an integer or number, such as the second argument of a `\setcounter` or `\addtocounter` command, or in

```
\hspace{\value{foo}\parindent}
```

It is useful for doing arithmetic with counters.

page 103ff

Putting the command `\extracolsep{width}` in an @-expression causes an extra *width* of space between all subsequent columns. This is in addition to the space normally put between the columns. Unlike the normal intercolumn space, this extra space between columns is not suppressed by an @-expression.

There is a new *-form of the `tabular` environment, which has an extra initial argument that specifies the width of the box that it produces. When using the `tabular*` environment, you should use the `\extracolsep` command in the argument to add some rubber space (such as that provided by `\fill`) to guarantee that the environment fills up the specified width. For example,

```
\begin{tabular*}{4in}[t]{ll@{\extracolsep{\fill}}
lr@{\extracolsep
```

```
{0in}\hspace{\tabcolsep}}r} ...
```

produces a five-column environment that is four inches wide. Extra space is added after the second and third columns to fill out the four inch width, while there is a standard intercolumn (`\tabcolsep`) space after the first and fourth columns.

page 104

Two `\hline` commands in a row produce two lines with a space between them. The rules produced by `-` characters in the `array` or `tabular` environment's argument are interrupted by this space.

pages 117-8

The following change is made obsolete in version 2.09.

Two new commands `\envbegin` and `\envend` have been added for use with the environment-defining commands introduced in version 2.07. A `\begin` or `\end` command cannot invoke an unmatched `\begin` or `\end` command. Instead, it must invoke `\envbegin` or `\envend`, as indicated in the following example.

```
WRONG: \newenvironment{foo}{\begin{list}...}
{\end{list}}
RIGHT: \newenvironment{foo}{\envbegin{list}...}
{\envend{list}}
```

page 121

The `\head` command has been eliminated. In its place are the commands `\markboth` and `\markright` having the format

```
\markboth{left head}{right head}
\markright{right head}
```

for setting either both or just the right heading. In addition to their use with the `myheadings` page style, you can use them to override the normal headings in the `headings` style, since `LATEX` uses these same commands to generate those heads. You should note that a left-hand heading is generated by the last `\markboth` command before the end of the page, while a right-hand heading is generated by the first `\markboth` or `\markright` that comes on the page if there is one, otherwise by the last one before the page.

page 137

The `\cite` command now has an optional argument that adds a note to the citation. For example,

```
\cite[pages 112-134]{jones:foo}
```

might produce “[34, pages 112-134]”. This obviously shouldn’t be used with a `\cite` command that produces multiple citations like “[12,34,45]”.

Changes in Version 2.09

no relevant page

The `\envbegin` and `\envend` commands introduced in Version 2.07 have been eliminated. The arguments of `\newenvironment` can contain `\begin` and `\end` commands.

no relevant page

A new `\lefteqn` command has been added, mainly for use in an `eqnarray` or `eqnarray*` environment (introduced in version 2.05), to help format wide equations that need to be broken across lines. It typesets its argument in display math style, but treats it as if it had zero width when aligning columns. For example, you can produce

by typing

```
\begin{eqnarray*}
\lefteqn{\sum_{1}^{N} u_{i}+v_{i}+w_{i}=\}\} \\
& \& a + b + c + d + e + f + g + h + i \\
& \& \mbox{} + j + k + l + m + n + o + p
\end{eqnarray*}
```

(The `\mbox{}` tells $\text{T}_{\text{E}}\text{X}$ that the following `+` is a binary rather than a unary operator.)

page 28, top

Eliminate the six lines beginning with `\hbox(6.94444`. All such printout has been eliminated from overfull and underfull box messages. (It will, however, appear in the log file.) This and all other box descriptions can be made to appear on the terminal with the `\showoverfull` declaration. (It obeys the normal scoping rules.)

page 39

Theorem-like environments now have an optional argument. If a lemma environment has been defined with a `\newtheorem` command, then typing

```
\begin{lemma}[Zorn] For every ...
```

will produce something like

Lemma 4 (Zorn) For every ...

page 95, middle

The `\sloppy` command has been modified so it never produces an overfull box. (Well, hardly ever.) There is also a `sloppypar` environment that typesets one or more paragraphs with `\sloppy` in effect. (The environment starts and ends a paragraph, so you don't have to worry about leaving a blank line at the end as you do when using the `\sloppy` declaration.)

page 97

There are many places, such as before and after displayed equations or list environments, where the `\nopagebreak` command has no effect. To overcome this problem, a new `\samepage` declaration has been added. It inhibits page breaking almost anywhere within its scope except between paragraphs, where `\nopagebreak` works nicely, or where a `\pagebreak` or `\nopagebreak` command explicitly allows breaking. The rules determining where it applies are a bit complicated; to a first approximation, it applies to an entire paragraph if its scope includes the blank line ending the paragraph.

page 137

A `\nocite` command has been added. Like `\cite`, it tells Bib_TE_X to add one or more references to the reference list, but it puts nothing in the text. It must come after the `\begin{document}` command.