

THE ARTIFICIAL INTELLIGENCE LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Working Paper 273

May 1985
Revised, June 1986

How to Use γ TeX

Daniel Brotsky
June 9, 1986

γ TeX—pronounced *why-TeX* or *oops-TeX*—is a TeX macro package. γ TeX provides both an easy-to-use interface for TeX novices and a powerful macro-creation library for TeX programmers. It is this two-tier structure that makes γ TeX more useful to a diverse TeX user community than other macro packages such as Plain or \LaTeX .

This paper contains γ TeX instructions intended for novice users. It summarizes the facilities provided in γ TeX and concludes with a table of useful commands.

The version of γ TeX documented here is release 2.0.

Work on γ TeX was supported by a desire to avoid doing real work, like research.

A.I. Laboratory Working Papers are produced for internal circulation, and may contain information that is, for example, too preliminary or too detailed for formal publication. It is not intended that they should be considered papers to which reference can be made in the literature.

What is Υ TeX?

Υ TeX—usually pronounced *why*-TeX but see below—is a TeX macro package. Υ TeX evolved from the author’s experience with other TeX macro packages, primarily TBASE, Plain, L^ATeX, and the PHW book macros. This experience showed that, on the one hand, packages as complete as L^ATeX or TBASE tend to be inflexible while, on the other hand, packages as flexible as Plain tend to be incomplete. Υ TeX ties both hands together by aiming for inflexibility and incompleteness simultaneously.

Like most TeX-related programs and macro packages, Υ TeX has a “cute” name that attempts a dual-language pun in English and pseudo-Greek.* The letter preceding the TeX in Υ TeX is not just a ‘Y’ but also an *upsilon*—the Greek antecedent for the English letters ‘u’ and ‘y’.† Thus you may pronounce Υ TeX as the expected *why*-TeX, the erudite *upsilon*-TeX, or the shorter *oops*-TeX. But when you spell it to the computer, you have to spell it Υ TEX.

About This Document

This document contains basic instructions for using Υ TeX. It describes the things you can do using the user interface that Υ TeX always provides. It does *not* explain how to use the Υ TeX library facilities to add to and replace parts of that interface. Readers who are interested in such matters should look at the Υ TeX sources.

If you are a complete TeX beginner you may find this document hard to follow. In fact, almost everyone will find parts of the presentation hard to follow. My advice to those who are confused is to not worry about it: most of the time you will be confused because you are reading about a feature you have no use for and thus no experience with. If you need to use something and don’t understand what is said about it here, just try the examples I give in one of your manuscripts and see if you can figure it out from there. If that doesn’t work, try finding someone who does use the feature and ask them. As a last resort, try finding a TeX wizard and asking them.

*The result is about as weak the premise suggests.

†Experienced TeX math hackers may know that TeX makes a version of capital upsilon available as Υ , and they may wonder why the name Υ TeX is not spelled as Υ TeX. That’s because TeX’s version of Υ is not a Greek version (one of which really looked like ‘Y’—trust the Romans to be accurate in their borrowing) but actually the Uncial version popular among middle-ages religious circles. Hey, what can I say? Trust TeX to be medieval whenever possible.

If you wish a quick summary of most of the \TeX commands, you should turn to the tables of commands, parameters, and switches at the very end of this document.

Some Notation and Conventions I Use Throughout

In what follows, I try to use notation and conventions very similar to those Knuth uses in the \TeX book. For convenience,* I summarize most of the conventions here.

- Text set in typewriter type is suitable for \TeX input.
- Commands—also called *control sequences* or *macros*—all start with a backslash. *Alphabetic commands* such as `\section` and `\it` have one or more letters after their backslash, while *non-alphabetic commands* such as `\'` and `\&` have a single non-letter after their backslash. When you use an alphabetic command, you must have a space or other non-letter right after the command name (as in `{\it italics}`), and any spaces after the command are ignored by \TeX . When you use a non-alphabetic command, you can have anything at all after the command name (as in the Spanish `m\'agico`), and spaces you put there are *not* ignored by \TeX .
- Commands sometimes need *arguments*, such as the stuff typed in braces in

```
\section {This is the Argument.}
```

Most commands which use arguments—a process also called *taking* or *reading* arguments—expect their arguments to be enclosed in braces (as they were in this example). But there are other ways of specifying arguments, and these are detailed in what follows.

- \TeX and \TeX both have the notion of *parameters* which have *values*. For example, the value of the `\hsize` parameter is used as the normal length of lines in paragraphs. To set a parameter, you give the name of the parameter, an optional '=' sign, and then the desired new value for the parameter. For example, to set the normal line length to 5 inches, you could say either of the following things:

```
\hsize=5in
\hsize 5in
```

- There are several kinds of parameters: `<integer>` parameters hold integers (such as 3 or -1), `<dimen>` parameters hold distances (such as 2in or 5pt), `<glue>` parameters hold variable distances (such as 6pt plus 2pt minus 1pt), and `<toks>` parameters hold token lists (such as `{\bigsize\bf}`).†

* (and also for those many of us who find the \TeX book completely confusing)

† There are also `<muglue>` parameters, but only math wizards worry about those.

The different kinds of parameters are used for different purposes; for example, the `\hsize` parameter mentioned above is a \langle dimen \rangle because line length is a distance, and the `\pageno` parameter that holds the current page number is—as expected—an \langle integer \rangle parameter. But no matter what their type, you assign values to parameters in the manner described above.

- If you ever want to examine the value of a parameter—for example, if you want to know what it is so you can change it appropriately—you can get \yTeX to show you the value by putting `\showthe` in front of the parameter name. For example, if you said `\showthe\parindent` in your input file, \yTeX would print

```
>20.0pt.
... \showthe\parindent
```

on your terminal screen when it processes your input. It would then pause and wait for you to type a carriage return before continuing.

Format of a \yTeX Input File

\yTeX input is just normal \TeX input.* It is customary, however, to start \yTeX input files with a comment indicating that \yTeX is expected to process the file. For example, the manuscript file for this document starts with the line

```
% for yTeX
```

so anyone who looks at the file will realize this.

The first command in every \yTeX manuscript should be a `\typesize` command specifying the desired type size for the text of the document. For example, after the comment shown above, the manuscript for this document has the line

```
\typesize=11pt
```

As you can see, a `\typesize` command looks like a command for setting one of \TeX 's \langle dimen \rangle parameters, such as `\parindent`. However, the dimension that follows the equal sign must be one of 10pt, 11pt, or 12pt, since these are the only sizes that \yTeX supports for normal text.

If you forget to put a `\typesize` command at the beginning of your manuscript, \yTeX will complain with the error message

```
! You never gave a \typesize command.
```

Don't panic; just type a carriage return and \yTeX will use 10 point type. But you should add a `\typesize` command before you run the file again.

*Of course, this means $\TeX82!$ \TeX input, for example, will not work in \yTeX .

You can't give more than one `\typesize` command in a single manuscript. This is because $\Upsilon\text{T}_{\text{E}}\text{X}$ implements your desired size by magnifying a 10pt document by the right amount, and $\text{T}_{\text{E}}\text{X}$ does not like to change magnification in the middle of a run. This may seem like a drawback, but it's really a feature, because it means that the line and page breaks will not change if you run a manuscript through $\Upsilon\text{T}_{\text{E}}\text{X}$ a second time using a different typesize! So you can proof your document in one size and run it off in a different size without worrying about formatting changes.

How To Invoke $\Upsilon\text{T}_{\text{E}}\text{X}$

To invoke $\Upsilon\text{T}_{\text{E}}\text{X}$ on `0Z` you give the command `YTEX` to the `EXEC.*` $\Upsilon\text{T}_{\text{E}}\text{X}$ starts up by printing a release number and reading in any recent fixes to itself. It then looks for two files called `YMATH.TEX` and `YLOCAL.TEX` in your connected directory (and on `TEXINPUTS:` if they are not found locally). You can customize your $\Upsilon\text{T}_{\text{E}}\text{X}$ by putting appropriate commands in these two files.

The `YMATH.TEX` files are intended for favorite mathematics macros while the `YLOCAL.TEX` macros are for macros or parameter settings of any kind. Of course, all kinds of $\text{T}_{\text{E}}\text{X}$ commands can be put in either of these files, but keep the following in mind: separating math macros from others can make the job of merging different authors' manuscripts a lot easier.

Where are the Sources

The source for $\Upsilon\text{T}_{\text{E}}\text{X}$ lives in the directory `KS:(TEX.YTEX)` on `0Z` and is split primarily among the files `YTEX.MAC`, `YBASE.MAC`, `YFONTS.MAC`, and `YUSER.MAC`. The file `YLOG.MAC` contains a log of essentially all the changes made to $\Upsilon\text{T}_{\text{E}}\text{X}$ since its early stages; the first few lines of this file declare the current version number. The file `YSITE.MAC` contains definitions and parameter settings ("site changes") appropriate to the local installation. The file `YUSAGE.TEX` contains the source for this document.

The file `KS:(TEX.YTEX)YFIX.MAC` is where $\Upsilon\text{T}_{\text{E}}\text{X}$ looks for revisions made to the macros since the last version was dumped. Finally, an archive of correspondence about $\Upsilon\text{T}_{\text{E}}\text{X}$ can be found in the file `KS:(TEX.YTEX)YTEX.MAIL` on `0Z`.

*The commands `YTEXHE` and `YTEXTB` also exist for those who prefer the Helvetica or Times Roman font families.

The \TeX View of Documents

\TeX believes that documents are made up of chunks of text called *elements* which are laid out on pages. For example, each paragraph is an element, as is each chapter or section heading, each figure, each footnote, and so on. Many of the \TeX commands you will use most often serve to tell \TeX which text belongs to which element. For example, the command `\footnote` tells \TeX that its argument is a footnote element.

As \TeX encounters elements of various types in its input, it first tries to format the content of each element in the desired way. For example, when it finishes reading the contents of a paragraph, it first tries to break the contents into lines according to the rules currently in effect for paragraph elements. When \TeX finishes reading the text of a table, it tries to format that text according to the current rules for table elements.

Once \TeX accumulates enough elements to fit on a page, it tries to lay those elements out on a page according to the layout rules that apply to each element. For example, header elements appear as a single line alone at the very top of a page, while footnote elements appear tacked onto the bottoms of pages.

Page layout is the hardest part of \TeX 's job. This is because the rules for laying out various elements may come into conflict. For example, the rules for figures elements that appear in text may say that the figure should be separated from the surrounding text by one pica.* But a figure may be placed just before a section heading, and the rules for section heading elements may say that they should be preceded by two picas. In cases like this, \TeX does the best it can, but it may not do exactly what you want. So if you are very picky about the page layout of a particular document, you may have to tell \TeX exactly what you want at each place a conflict like this arises. But don't worry: usually only people who are publishing books need to be this picky.

The Five Kinds of \TeX Commands

There are five kinds of \TeX commands:

- *Delimiters* are commands that demarcate the text of particular elements: they do what is commonly known as "typemarking." For example, `\section` and `\footnote` are both delimiters. All delimiters read arguments (in ways explained below); it is the contents of these arguments that become the contents of the elements they delimit.

*There are six picas to the inch.

- *Parameters* are commands that affect the formatting and layout rules for particular elements. For example, `\hsize` and `\parskip` are parameters that affect the width and layout of paragraphs and pages. Parameters don't take arguments; instead, they have values which can be assigned in the manner described above.
- *Switches* are commands like parameters that affect the formatting and layout of particular elements. For example, `\noindent` and `\offheaders` are switches. Switches come in two kinds—on/off switches and true/false switches—which are described in detail later. Switches never take arguments.
- *Abbreviations* are commands that expand into common sequences of low-level \TeX commands. For example, `\padline` and `\linebreak` are abbreviations. Some abbreviations have arguments, and some don't. Any abbreviations that take arguments take them in braces in the normal \TeX way.
- *Descriptors* are commands that create new element types with particular formatting and layout rules. For example, the commands `\newfloat` and `\newtitle` are descriptors that are used to create elements which float to the top of pages or which title things. Descriptors take very complex arguments.

Normally, you use only delimiters, parameters, switches, and abbreviations in your manuscripts. Descriptors are for compulsive \TeX hackers who feel the need for new element types, or for wizards who are preparing documents with special requirements (such as this manual).

More About Delimiter Commands

Delimiters come in three kinds:

- *Standard* delimiters—such as `\section`, `\footnote`, or `\spread`—take their arguments in the standard \TeX way: the arguments are enclosed in braces and appear immediately after the command. For example, this is how you give two arguments to the `\spread` command:

```
\spread {First argument}{Second argument}
```

- *Paired* delimiters—such as `\begintext/\endtext` and `\beginabstract/\endabstract`—surround their argument. For example, this is how you delimit the text of an abstract:

```
\beginabstract
This is a very short abstract
used only for expository purposes.
```

```
I wish all my abstracts were this simple.
\endabstract
```

- *Paragraph* delimiters—such as `\bpar` and `\ftpar`—are kind of a cross between standard and paired delimiters. Their most important argument—the text of the paragraph—is delimited by the paragraph delimiter in front and the command `\par` (or a blank line) in back. But they may have additional arguments (such as tag text) which are given in braces immediately following the paragraph delimiter. For example, this paragraph starts with

```
\bpar {\it Paragraph\} delimiters---
```

and ends with a blank line. But it would look exactly the same if it had started with

```
\ftpar{${\bullet}$} {\it Paragraph\} delimiters---
```

instead.

Most paired delimiters are also available in a standard form. For example, you can use the command `\abstract` instead of `\beginabstract` and `\endabstract`, in which case you would enclose the text of the abstract in braces and place it right after the command.* As you may have guessed from the example, you get the standard form of paired delimiters by removing the begin and end prefixes.

Each argument to a standard or paired delimiter is a local group; that is, font and other parameter changes inside the element do not have any effects outside. So if you want the text of your abstract to be in italics, you can just say

```
\beginabstract
\it This abstract is in italics.
\endabstract
```

instead of the (also correct but) tedious

```
\beginabstract
{\it This abstract is in italics.}
\endabstract
```

Similarly, you can just say

```
\section {\it An Italic Section Header}
```

*The only paired delimiters that are not available in standard form are those in which the delimited text is in “nofill” mode, such as `\beginverbatim` and `\endverbatim`. This has to do with subtle differences between the effect of the two forms, differences that you needn’t care about and which can confuse even \TeX wizards.

instead of

```
\section {\it An Italic Section Header}
```

More About Switches

Switches come in two kinds:

- *On/off* switches are usually used to control whether things such as headers and footers appear on a page. For example, whether headers appear or not is controlled by the two commands `\onheaders` and `\offheaders`. These two commands together are said to control the `\headers` switch.

All on/off switches have variations in a *yes/no* form. For example, the commands `\yesheaders` and `\noheaders` are like `\onheaders` and `\offheaders`, except that they *only* control headers on the current page, not on future pages as well.

The point of having both on/off and yes/no versions of switches can be seen from the following: If you want none of the pages of a letter numbered, you would put `\offfooters` at the beginning of the letter. But if you want numbers on the bottoms of all pages except the first, you would put `\nofooters` at the beginning. The effect of `\nofooters` is to turn off the `\footers` switch for the first page *only*.

- *True/false* switches are mostly used by wizards, so I won't say much about them here. They are much like off/on switches except that they have a different form. For example, the `\vpar` switch controls whether the first paragraph after a section heading is left-flush or not; the `\vpar` switch is turned on with `\vpartrue` and turned off with `\vparfalse`. You can see the effects of saying `\vpartrue` in this document.

A List of the \TeX User Macros

This section contains descriptions of the most useful \TeX macros. The listing is broken down by subject area; for example, all the commands relevant to footnotes can be found in one place, and all the commands relevant to sectioning in another. The next section gives an alphabetized list of all the commands.

In the following descriptions, only the `begin` half of paired delimiters is mentioned. All the paired delimiters are available in standard forms unless the description mentions otherwise. Also, only the on form of on/off switches is mentioned. Of course, the off, yes, and no forms are also available.

Titling Pages

\TeX provides two kinds of titling pages: *part pages* which are delimited by `\beginpartpage` and *title pages* which are delimited by `\begintitlepage`. The only difference between the two is that title pages are always numbered 0 while part pages have whatever number they would normally be assigned. The contents of titling pages are centered by default, and titling pages have no runners unless you ask for them.

The following commands are useful both on titling pages and elsewhere:

`\begintitle` and `\beginctitle`

set their arg as titles. `\begintitle` gives a left-flush ragged-right paragraph and `\beginctitle` gives a ragged-center paragraph. Inside of titles, the commands `\\`, `\cr`, and `\crcr` force line breaks.

`\beginauthor` and `\begincauthor`

are like `\begintitle` and `\beginctitle` but use `\authorfont` (default `\regsize\rm`) instead of `\titlefont` (default `\bigsize\bf`).

`\beginabstract`

delimits abstract text which is set in `\abstractfont` (default `\smallsize\rm`) and indented from both margins by `\abstractindent` (default Opt).

The following command is available only on titling pages:

`\beginbottomtext`

delimits text which is set at the bottom of the page in `\bottomtextfont` (default `\smallsize\rm`). If you use this command, the text of the page will no longer be centered. Instead, the delimited material appears at the bottom of the page and material above it will start at the top of the page.

Chapters and Sections

These macros produce titles at the top of chapters, sections, and so on. Titles are normally set as left-flush, ragged-right paragraphs. Inside of titles, the commands `\\`, `\cr`, and `\crcr` force line breaks.

`\beginchapter <title> \endchapter`

starts a new chapter. The `<title>` is set in the `\chapterfont` (default `\bigsize\bf`) as a left-flush ragged-right paragraph. The title is preceded by `\prechapterpenalty` and `\abovechapterskip` and followed by `\postchapterpenalty` and `\belowchapterskip`. Most people prefer the standard form `\chapter {<title>}`.

`\beginpchapter`

is like `\beginchapter` but it forces the title to the top of a new page and turns off runners for that page. In two-sided matter, it forces the title to the top of a recto page.

`\beginsection`

is like `\beginchapter` but uses smaller font and spacing. Neither chapters nor sections do any automatic numbering for you. But there are no-op commands `\secdef` and `\secref` which, like their figure and table cousins, can be used to help keep track of numbering. There are commands `\beginsubsection` and `\beginsubsubsection` but they are just aliases for `\section`.

`\beginheading`

is like `\beginsection` but with smaller font and spacing.

The true/false switch `\centerheadingstrue` controls whether chapter, section, and other titles are set as left-flush or centered ragged-margin paragraphs. The default is `\centerheadingsfalse`; that is, headings are left-flush as mentioned above.

Page Layout

Pages have headers and footers, jointly called runners. When they are turned on, runners are single lines that appear separated from the main text by the distances `\headerdrop` and `\footerdrop`.^{*} By default, headers are turned on and footers are turned off.

`\onrunners`, `\onheaders`, and `\onfooters`

are switches that turn the various runners on. Recall that the effects of these commands are permanent; the effects of their yes/no counterparts are confined to the current page.

There are actually two versions of headers and footers—one for recto (right-hand or odd-numbered) pages and one for verso (left-hand or even-numbered) pages. By default, all pages are considered to be recto pages, but you can use the switch `\twosidedtrue` to get two-sided output and `\twosidedfalse` to go back to one-sided.

The contents of headers and footers are the values of the `(toks)` parameters `\versoleftheader`, `\rectoleftheader`, and so on. By default, the `\rectorighthead`, the `\versolefthead`, and both center footers contain a boldface page number (called a *folio*); these come out in roman numerals

^{*}These dimensions are not measured baseline-to-baseline; rather, they are the appearing space between the runners and the text.

if they are negative. (The page number is the parameter `\pageno` which you can set.) Also by default, the `\rectolefthead` is the `\firstmark`.

For example, if you want your pages numbered at the bottom, you could say

```
\offheaders    % turn off the header lines
\onfooters     % turn on the footer lines
```

because the footers contain page numbers by default. If you want each page to have a title and a page number, you can say

```
\rectolefthead={The Title of My Paper}
```

Since this is done so often, there is a synonym `\runninghead` for `\rectolefthead` that lets you say

```
\runninghead={The Title of My Paper}
```

instead.

The default font for runners is kept in the (toks) parameter `\runnerfont`, which starts out as `\smlsize\rm`.

Paragraphs

Each paragraph is an element to \TeX . Paragraphs are considered to consist of a crown (the first line) and a vest (all the other lines). By default, the crown is indented `\parindent` and the vest is not. But there are a variety of paragraph delimiters that give you paragraph elements that are formatted differently. (Recall that paragraph delimiters start paragraphs that are ended by `\par` or a blank line.)

`\ivpar`

delimits an *inverted* paragraph: the crown is not indented but the vest is.

`\ipar`

delimits an *indented* paragraph: both crown and vest are indented.

`\ftpar {<tag text>}`

delimits a *flush-tagged* paragraph: both crown and vest are indented and the first argument (in braces) to `\ftpar` is set flush on the left margin of the crown. For example,

```
\ftpar {1.} First line ...\linebreak second line ...\par
produces a paragraph that looks like this:
```

1. First line ...
second line ...

`\atpar {<tag text>}`

delimits an *adjoint-tagged* paragraph: both crown and vest are indented

and the first argument (in braces) to `\atpar` is set so its right edge is `\partagsep` away from the left edge of the crown line. (This is like the Plain \TeX `\item` command.) For example,

```
\atpar {a)} First line ... \linebreak second line ... \par
```

produces a paragraph that looks like this:

```
a) First line ...
   second line ...
```

`\vtpar {<tag text>}`

delimits a *variably-tagged* paragraph: the crown, vest, and tag are set just like an adjoint-tagged paragraph but the width of the indent is not the `\parindent` but instead is the width of the tag text. For example,

```
\vtpar {Keywords:} This ... \linebreak and that ... \par
```

produces a paragraph that looks like this:

```
Keywords: This ...
           and that ...
```

`\bpar`

delimits a *bullet* paragraph: a flush-tagged paragraph whose tag is a bullet symbol.

If bullet paragraphs appear next to each other, it is good practice to put a `\beginbullets` before the first one and an `\endbullets` after the last one. This will add a little space before and after so as to visually cluster the paragraphs.

If you interpose math or a figure or a title between paragraphs, it is standard typesetting practice not to indent the paragraph immediately following the interposed element. For example, the paragraphs which immediately follow section titles in this document are not indented, nor are those which follow figures or textual displays.

To follow this practice in your own documents, you could use `\noindent` to start the paragraphs which follow interposed elements. But \TeX does this for you automatically. The command `\vpar` effectively forces the next paragraph to start with `\noindent`, and all \TeX commands which interpose material between paragraphs—such as `\section`—end with `\vpar`. Thus you will not have to use `\noindent` after figures and so on.

If for some reason you want a particular paragraph that follows interposed text to be indented normally, just give the command `\unvpar` in front of that paragraph. `\unvpar` cancels the effect of `\vpar` and allows the paragraph to be indented.

\vpartrue and \vparfalse

allow and disallow the effects of \vpar . If you don't want the first paragraphs of sections and so on to be left-flush, say \vparfalse .

 \linebreak

ends the current line of a paragraph, filling out to the right hand margin with blank space.

 \onindent

turns the default indentation of crown lines on. If you want your paragraphs to normally have no indentation, say \offindent . But keep in mind that this will break commands like \bpar which rely on indentation.

 \beginquote

delimits a quote. Quotes are set in the $\quote\font$ (default $\small\size\rm$) with the left and right margins narrowed by \parindent . Inside of quotes paragraph indentation is turned off (with \offindent).

Fonts and Sizes

To change type face in a document, you use the normal \TeX commands \it , \bf , \tt , \sl , and \rm . To change to increasingly bigger sizes of type, you can use the commands $\big\size$, $\bigg\size$, and $\biggg\size$. For increasingly smaller sizes, you can use $\small\size$, $\smalll\size$, and $\smallll\size$. You can switch to the regular size with $\reg\size$. The size-changing commands all switch to the roman face.

There are commands $\small\rm$, $\smalll\rm$, and $\small\tt$ which are like face-changing commands in that the baseline spacing does not change, but the "typeface" they switch to are actually smaller sizes of the indicated face. For example, to get the effect of what printers call "small caps", you can use the $\smalll\rm$ face. To get LISP, you would say $L\{\smalll\rm\ ISP\}$.

If you ever want to set material in the font selected by a font parameter such as $\title\font$, you can select that font by saying \the in front of the parameter name. For example, to set three words in the $\caption\font$, I could say

Here are $\{\the\caption\font\ three\ small\ words\}$.

which gives me

Here are three small words.

The command $\single\space$ causes lines to be single spaced, and the command $\double\space$ causes lines to be double spaced. Single spaced lines are the default. These commands affect all text, including footnotes and captions, so that if you use $\double\space$ you might want to add a $\single\space$ command to the font parameters for footnotes, captions, and so on. If you don't

like the exact spacing values used by `\singlespace` or `\doublespace`, read what it says below about line spacing for wizards and define your own versions of these commands.

Figures

You can ask \TeX to lay out your figures in one of four ways: *stationary*, which means that figures are boxes that appear where they are defined; *section*, which are like stationary except that they are followed by `\vfil` glue which makes them suitable for putting all in one section with no intervening text; *top*, which means all figures float to the top of a page; or *floating*, which means they appear where they are defined unless they can't fit on the page there in which case they float to the top of the next page they fit on. The current layout discipline (default *floating*) can be set with `\stationaryfigures`, `\sectionfigures`, `\topfigures`, and `\floatingfigures`.

`\figdef` and `\figref`

are no-ops which are useful for keeping track of figure numbers.

`\beginfigure`

delimits a figure definition. You get a figure layed out according to the current discipline. Each figure is enclosed in a `\vbox` so you don't have to worry about breaks.

This figure was defined with

```
\beginfigure
\noindent This figure was defined with ...
...
\begincaption
Figure~\figdef{1}. A figure which is ...
... it will float.
\endcaption
\endfigure
```

and this was the result.

Figure 1. A figure which is also an example of how to prepare a figure. This figure will appear where it was defined unless it can't fit on that page, in which case it will float.

`\beginpagefigure`

delimits a full-page figure definition. These figures go on pages by themselves.

This full-page figure was defined with

```
\beginpagefigure
\vfil
\noindent This full-page figure was defined with ...
...
\begincaption
Figure~\figdef{2}. A figure which is ...
... it will float.
\vfil
\endcaption
\endpagefigure
```

and this was the result.

Figure 2. A figure which is also an example of how to prepare a figure. This figure will appear where it was defined unless it can't fit on that page, in which case it will float.

\backslash beginstationaryfigure

gives a stationary figure no matter what the current layout discipline is. The paired delimiter \backslash beginstationarypagefigure gives a full-page stationary figure, and there are similar explicit commands provided for all four of the layout disciplines.

 \backslash begincaption

is defined only within figures and delimits captions, which are set as unindented paragraphs in \backslash captionfont with margins narrowed by \backslash captionindent. Captions are preceded by \backslash abovecaptionskip.

For examples, see figures 1 and 2, which I referred to here by saying

... see figures \backslash figref{1} and \backslash figref{2}, ...

Note, in particular, that figure 1 is bracketed by horizontal lines. \TeX will put out such lines if you say \backslash figurelinetrue at the front of your document; the default is \backslash figurelinefalse (no lines). (The lines themselves are put \backslash figurelinedrop away from the body of the figure.)

The easiest way to leave space for a paste-in figure is to define a figure whose body consists of a vertical skip command, as in

```
 $\backslash$ beginfigure
 $\backslash$ vskip 2 true in
 $\backslash$ begincaption
This will be a 2 inch high pasted in figure.
 $\backslash$ endcaption
 $\backslash$ endfigure
```

Note that the amount to skip was given in true inches, not just inches. This is because typesizes larger than ten point are implemented by magnifying the entire document: if the dimension had been given as 2in and the typesize were given as 11pt, \TeX would leave 2.2 inches of space instead of 2. The use of true in the dimension specification prevents \TeX from magnifying the space it leaves for the paste-in material.*

There is a \langle toks \rangle parameter \backslash topsep whose contents are inserted between the floating figures and text on any page where floating figures appear. By default, \backslash topsep is empty, so only the normal below-figure space appears between floating figures and text. If you want more space, you can say, for

*Keep in mind that specifying sizes in true dimensions may cause the page breaks to change when the type size changes. If you are proofing in a larger typesize and you want page breaks to remain consistent, you should specify paste-in sizes *without* the true specification, basing the declared size on the magnification which will be used in *final* output. Then don't worry about the fact that the spaces in the proof version are a little large.

example,

```
\topsep={\bigskip}
```

Tables

Tables are delimited with `\begin{table}` and `\end{table}`. Right after the `\begin{table}` command you must have a *preamble specification* enclosed in brackets ([and]), such as in

```
\begin{table} [l"r|lc]
\topline
Item&&      Price&&  User&      Use\cr
\dmidline
Widgets&&    \$2.50&&  D. Brotsky&  \TeX\ work\cr
\midline
Grommets&&\$2500.50&& The Pentagon& none\cr
\botline
\end{table}
```

which produces table 1.

Item	Price	User	Use
Widgets	\$2.50	D. Brotsky	\TeX work
Grommets	\$2500.50	The Pentagon	none

Table 1. This table is produced by the example in the text. Well, actually, this table is *almost* produced by that example; in fact, this one has also been centered on the page by putting the entire text of that example into the argument of `\centerline`.

Inside of a preamble specification, each character stands for one column in the table. The meanings of the characters are:

- l left-aligned column
- L left-aligned math column
- c center-aligned column
- C center-aligned math column.
- r right-aligned column
- R right-aligned math column
- | single-thickness vertical line
- " double-thickness vertical line
- & repeat specification marker

Math columns are just like their non-math counterparts except that the entries are set in math mode in `\tablestyle` (default `\displaystyle`). The `&`

specification allows you to take advantage of \TeX 's automatic iteration of column specifications: if you have more entries in a line than you have columns in the preamble, the preamble columns will repeat from wherever the `&` was specified.

The thickness of a `|` line is the \langle dimen \rangle parameter `\vbarwidth` (default `.4pt`) and the thickness of a `"` line is the \langle dimen \rangle parameter `\dvbarwidth` (default `1pt`). Keep in mind that `|` and `"` columns must be delimited by `&` like other columns. It's just that normally the contents of these columns are empty, so you end up with `&&` in the rows of your input.

Inside tables the command `\\` ends rows like `\cr`. In addition, you get access to a variety of useful commands:

`\inline`

puts a horizontal line in the table. Better to use are `\topline`, `\midline`, and `\botline` (which you normally use at the top, middle, and bottom of the table) because they put padding around the line which looks good.

`\dinline`, `\dtopline`, `\dmidline`, and `\dbotline`

are the double-width counterparts of the commands mentioned above. `\inline`-type lines have thickness `\hbarheight`, and `\dinline`-type lines have thickness `\dhbarheight`.

`\padline` \langle dimen \rangle

puts \langle dimen \rangle extra space between lines of the table with the nice feature that vertical bars specified in the preamble extend right through the extra space. The \langle dimen \rangle is not an argument: don't put it in braces.

In addition, you get `\tabdef` and `\tabref` which, like `\figdef` and `\figref`, are no-ops useful for keeping track of table numbers. Also, there are the following table-affecting parameters:

`\pretabskip`, `\intabskip`, and `\posttabskip`

are the glue parameters that `\tabskip` glue is set to before the first column, between columns, and after the last column of a table. Defaults are `\hfil` glue for `\pretabskip` and `\posttabskip`, `1em` glue for `\intabskip`.

`\tablewidth`

is settable like a \langle dimen \rangle , and setting it has strong side effects. If you set it positive, all lines in tables are set to that width. If you set it to zero, tables have their natural width. If you set it negative, tables are expanded by the negative of what you set it to. The default is `0pt`; that is, tables have their natural size.

Finally, in addition to the `\beginable` command which puts a box around its table, there is `\beginopentable` which does not put a box and there is `\begindisplaytable` which centers the whole table in a $$$$ -style display.

Footnotes

\TeX gives you a footnote mechanism essentially equivalent to Plain's, but footnotes are output as adjoint-tagged paragraphs in the `\footnotefont` with an indent of `\footnotemarkwidth`.

`\footnote{<tag>}{<footnote text>}`

puts `<tag>` where you call `\footnote` and also gives you a footnote marked with `<tag>`. You use `\footnote` inside paragraphs.

`\nfootnote{<tag>}{<footnote text>}`

is like `\footnote` but the `<tag>` must be a number. The tag is set as a superscript.

`\vfootnote{<tag>}{<footnote text>}`

is like `\footnote` but it doesn't put the tag where you call it, only on the footnote text itself. You use `\vfootnote` between paragraphs.

`\beginfootmatter`

gives you a footnote with no marker and with no left margin indentation where the marker would go. I'm not sure why you would want this.

The no-ops `\footdef` and `\footref` are provided to help keep track of footnote numbering. There is a `<toks>` parameter `\botsep` whose contents are inserted between the text and footnotes on any page where footnotes appear. By default, `\botsep` puts in some space and a short line. If you just want space, you can say, for example,

```
\botsep={\bigskip}
```

Textual Displays

These macros give you $$$$ -like displays containing text and other useful textual blocks. There are no standard forms for any of the begin and end paired delimiters that produce textual displays.

`\beginnofill`

delimits text that is not made into filled lines, but output in lines as it appears in the input. Indentation is turned off.

`\begintext`

gives you a $$$$ -like display containing the nofill material it delimits. The displayed lines have width `\displaywidth` and are indented `\parindent` from the normal `\displayindent` of the display.

\backslash begintextlines

is like \backslash begintext but the displayed lines are the full \backslash hsize wide and the normal \backslash displayindent is ignored.

For example, this is a paragraph which has both its crown and vest indented twice the normal paragraph indentation. If, inside this paragraph, we produce a display with

```
 $\backslash$ begintext
Here is the first line of the display
Here is the second.
 $\backslash$ endtext
```

then the result appears to have a left margin even with the indentation of the paragraph, as in:

```
Here is the first line of the display
Here is the second.
```

(Each line of the display is indented \backslash parindent from the margin because each line starts a paragraph.) But if we specify the text of the display using the following commands:

```
 $\backslash$ begintextlines
Here is the first line of the display
Here is the second.
 $\backslash$ endtextlines
```

then the display ignores the indentation of the paragraph, producing the following:

```
Here is the first line of the display
Here is the second.
```

 \backslash begincode and \backslash begincodelines

are like \backslash begintext and \backslash begintextlines but the material is set in the typewriter font.

 \backslash verb

is used as

```
 $\backslash$ verb<char><text><char>
```

and gives you \langle text \rangle verbatim in the typewriter font. All \TeX special characters (such as \backslash and $\&$) are treated as normal characters.

 \backslash beginverb and \backslash beginverblines

are like \backslash begincode and \backslash begincodelines except the intervening text is taken verbatim.

For example

```
\beginverb
\noindent This is some \verbatim \text.
\centerline {Note that commands are ignored.}
\endverb
```

produces

```
\noindent This is some \verbatim \text.
\centerline {Note that commands are ignored.}
```

\backslash beginlisp and \backslash beginlisplines

delimit LISP programs. Programs are printed in the \backslash lisplfont (default \backslash regsize \backslash tt) in nofill mode; the \TeX control characters (such as #) often found in LISP code are turned off; and comments are printed in roman. Programs are boxed to prevent breaks; the sequence \backslash pbrk at the start of a line makes that line an escape into surrounding vertical mode: you can put in breaks and glue. If you just specify \backslash pbrk after the semicolon, with nothing else on the line, you get a space suitable for insertion between function definitions.

For example, this input

```
\beginlisp
;;Good Old Factorial
(defun fact (n)
  (cond ((zerop n) 1)           ;base case
        (t (* n (fact (1- n)))) ;recursive case
  )
;\pbrk
(defun myfact (n)
  (cond ((> n 0) (fact n))     ;OK if pos
        (t (- (fact (- n))))  ;invert if neg
  )
;\pbrk \medskip
(fact -5)                       ;sample usage
  -125
\endlisp
```

produces this output

```
;;Good Old Factorial
(defun fact (n)
  (cond ((zerop n) 1)           ;base case
        (t (* n (fact (1- n)))) ;recursive case
  )
(defun myfact (n)
  (cond ((> n 0) (fact n))     ;OK if pos
        (t (- (fact (- n))))  ;invert if neg
  )
```

```
(fact -5) ;sample usage
-125
```

Two-Column Text

The paired delimiter `\begintwocolumntext` sets its argument material in two columns, with the columns separated by `\columnskip`. Top figures and footnotes in this material are still set in a single column extending the full width of the page. Note that you can not use the floating figure style in two-column text because there is no way of knowing in advance whether figure material set in this style should be one- or two-columns wide. Also, full-page stationary or section figures are liable not to work.

The paired delimiter `\begindoublecolumn` sets its argument material two columns per page, just as if the `\hsize` had been narrowed and the results pasted up. Thus, every page becomes a single column, and commands like `\eject` refer to columns, not pages. The command `\pageeject` will force a true page break. *Warning:* The `\begindoublecolumn` command will throw away everything on the current page, so it is only safe to use at the start of a fresh page.

Formatting Small Pieces of Text

These commands produce little chunks of text in useful shapes, or combine small blocks of text in handy ways.

`\spread`

takes two or more arguments and spreads them out evenly spaced on a line. For example, the input

```
\spread{Piece1}{Piece2}{ }{Piece4 (3 was empty)}
```

produces

```
Piece1           Piece2           Piece4 (3 was empty)
```

Each piece of text except the leftmost and rightmost are centered on their appropriate position. Notice that the empty argument took up a position even though it didn't put anything in that place.

`\begintleft`, `\begintcenter`, and `\begintright`

set their arguments as left-flush, centered, and right-flush titles in the current font. Titles are allowed to contain more than one paragraph. Inside of titles, the commands `\\`, `\cr`, and `\crcr` force line breaks.

`\sy` and `\sybox`

set their arguments in a slightly smaller typewriter font. They differ only in that `\sybox` also puts the argument in an `hbox` so it can't be broken across lines.

`\ucsy` and `\ucsybox`

are relatives of `\sy` and `\sybox` that put their arguments in upper case.

`\ignore`

does just that to its single argument.

`\filpage`

fills up the rest of the page with white space and does a page break.

The Date

`\hour` and `\minute`

are `(number)` parameters that hold the time \TeX started in 24-hour format. `\daytime` outputs this in the form 15:35.

`\monthname` and `\monthshortname`

output the name of the current month; `\monthshortname` is three letters long.

`\shortyear`

names the current year without the first two digits, as in 86.

`\date`, `\shortdate`, and `\slashdate`

output the current date as in June 9, 1986; 9 Jun 86; and 6/9/86.

For example, you can get output like

Today is day 14 of Aug (*i.e.* August), and the time is 14:34.

by saying

```
Today is day \number\day\ of \monthshortname
({\it i.e.} \monthname), and the time is \daytime.
```

Miscellany

`\ytex`, `\yTeX`, `\YTEX`, `\YTEX`, and `\oopstex`

all give the \TeX symbol.

`\draft`

adds a line to the bottom of each page that looks like this:

DRAFT • DRAFT • DRAFT **June 9, 1986 [15:04]**

This command also sets the `\overfullrule` to 5pt instead of the default 0pt and prints the message `{Draft}` on your terminal screen.

Here are the default values of a few parameters not yet mentioned:

```
\hsize=28pc \vsize=44pc
\abovedisplayskip=3pt plus1pt minus2pt
\belowdisplayskip=3pt plus1pt minus2pt
\abovedisplayshortskip=0pt plus1pt
```



```

\belowdisplayshortskip=2pt plus1pt minus1pt
\parindent=2em
\parskip=0pt plus1pt
\normallineskip=2pt
\normallineskiplimit=0pt
\clubpenalty=900
\widowpenalty=900

```

The default `\typesize` is ten point. Keep in mind that, if you make the `\typesize` 11pt or 12pt, the `\hsize` and `\vsize` will be magnified by 1.1 or 1.2, so you probably won't need to change them. If you have a definite measurement you want used for the margins independent of what magnification gets used, specify it in "true" units (as in `\hsize=6 true in`). The output routine will always try to center your output on the output page; you can use `\hoffset` and `\voffset` to shift it off center any desired amount.

For Wizards Only

Here are some features that inexperienced \TeX nicians should probably avoid. Once again they are grouped by topic: some of the topics also appeared above.

Page Glosses

Pages can have *glosses* which are special lines that appear above the header and below the footer. The \langle toks \rangle parameters `\topgloss` and `\bottomgloss` contain the contents of the glosses. You turn glosses off and on just like runners, but both glosses default off. The top gloss defaultly contains a copyright message like the following:

```
Copyright © 1986 by the author 9 Jun 86 [15:04]
```

while the bottom gloss is defaultly empty. The words "by the author" are the value of the \langle toks \rangle parameter `\copyrightholder`.

```
\showcopyrighttrue and \showcopyrightfalse
```

allow and disallow the showing of the top gloss. To actually get the top gloss to show on a particular page, you also have to say `\copyrightpage` somewhere on that page.

The `\draft` macro described above uses the bottom gloss.

Size Changing

There are also explicit size-selection commands such as `\twelvepoint`, but these pay no attention to the `\typesize`, so their use is not advised.

The baseline spacing is set whenever a size change command is given by saying

```
\normalbaselineskip=\the\baselinefactor em%
\baselineskip\normalbaselineskip
```

Thus, the value of the \langle toks \rangle parameter `\baselinefactor` (default `{1.3}`) can be used to vary the baseline spacing. Note that changes in `\baselinefactor` will not take effect until you give a size-changing command or say `\setnormalbaselines`. For example, these are the default definitions for `\singlespace` and `\doublespace`:

```
\def\singlespace{\baselinefactor={1.3}\setnormalbaselines}
\def\doublespace{\baselinefactor={2.6}\setnormalbaselines}
```

Setup Hooks

A variety of the default user-level macros—such as figures, captions, footnotes, etc.—provide for user-defined set-up hooks. For example, every figure macro calls the macro `\setupfigure`, which in turn calls a macro `\setupfigurehook` which the user is free to define. Check the source code for details as to which of these hooks exist.

More About Tables

There is an escape mechanism whereby arbitrary alignment specifications can be put into the preamble of a table. A column is normally specified by a letter, but arbitrary material enclosed within balanced braces may be used to specify a column instead. For example, the preamble specification

```
[l{\hfil $\displaystyle #}$c]
```

produces the following preamble

```
\tabskip\intabskip
#\hfil & \hfil $\displaystyle #$$ \hfil #\hfil
\tabskip\posttabskip\cr
```

The material within matching braces can only specify one column in the preamble: you must use more than one set of braces to specify more than one column. There is no limit on the number of columns that can be specified in this way. When a column is specified in this way, the entry produced for that column by `\padline` will contain an `\omit` so as to produce white space. Thus, if the specified material contains a `\vrule` specification, that rule will not extend through padding.

If the material in braces starts with a `\tabskip` specification, \TeX assumes that it contains only that specification, and it merely embeds the specification in the preamble instead of making it a column specification. For

example, the specification

```
[l{\tabskip=2em }lr]
```

produces the preamble

```
\tabskip\intabskip
#\hfil & \tabskip=2em #\hfil& \hfil #%
\tabskip\posttabskip\cr
```

This mechanism can be used to vary the between-column skip within a line. Keep in mind, however, that the `\posttabskip` will override such specifications after the last column.

The interline glue is normally turned off in tables, and struts are used to keep the line spacing correct. If all of the preamble specifications in a table are given with the `{}` escape mechanism instead of \TeX 's abbreviations, you should put struts in at least one of the columns to force correct line spacing.

If you specify height \langle dimen \rangle as the contents of a `|` or `"` column, that will override the natural height of the `\vrule`. This is the mechanism used by `\padline`.

A List of Many \TeX Commands

Here is a list of all the commands, parameters, and switches described hereto. It is not a complete list of all the \TeX commands, but it gets most of the useful ones.

The Commands

The type given in this table is one of **SD** for standard delimiter, **PD** for paired delimiter, **Par** for paragraph delimiter, and **A** for abbreviation. If a paired delimiter has a standard form, it appears in this table in that form. Paired delimiters which do not have standard forms (*i.e.*, those which delimit nofill text) appear in their `begin` form alphabetized under `begin`.

Command	Type	Summary
<code>\\</code>	A	breaks lines in titles, ends rows in tables
<code>\abstract</code>	PD	abstract text
<code>\atpar</code>	Par	adjoint-tagged paragraph
<code>\author</code>	PD	author's name(s)
<code>\begincode</code>	PD	typewriter font textual display
<code>\beginodelines</code>	PD	unindented typewriter font textual display
<code>\beginlisp</code>	PD	LISP program display
<code>\beginlisplines</code>	PD	unindented LISP program display
<code>\beginnofill</code>	PD	unboxed unfilled text

<code>\begintext</code>	PD	unfilled textual display
<code>\begintextlines</code>	PD	unindented unfilled textual display
<code>\beginverb</code>	PD	typewriter font verbatim display
<code>\beginverblines</code>	PD	unindented typewriter font verbatim display
<code>\bigsize</code>	A	switch to a larger size of type
<code>\biggsize</code>	A	switch to an even larger size
<code>\biggsize</code>	A	switch to an even larger size still
<code>\botline</code>	A	horizontal line for bottom of tables
<code>\bottomtext</code>	PD	text for bottom of title page
<code>\bpar</code>	Par	bullet paragraph
<code>\bullets</code>	PD	group of bullet paragraphs
<code>\caption</code>	PD	caption (figures only)
<code>\cauthor</code>	PD	centered author's name
<code>\chapter</code>	PD	chapter title
<code>\copyrightpage</code>	A	allow copyright message
<code>\cr</code>	A	breaks lines in titles, ends rows in tables
<code>\crcr</code>	A	breaks lines in titles, ends rows in tables
<code>\ctitle</code>	PD	centered title
<code>\date</code>	A	date as June 9, 1986
<code>\daytime</code>	A	time of day in 24 hour format
<code>\dbotline</code>	A	double width line for bottom of table
<code>\dinline</code>	A	double width unpadded table line
<code>\displaytable</code>	PD	table display
<code>\dmidline</code>	A	double width line for middle of table
<code>\doublecolumn</code>	PD	double-column mode
<code>\doublespace</code>	A	double space text
<code>\draft</code>	A	put draft mark on all pages
<code>\dtopline</code>	A	double width line for top of table
<code>\figdef</code>	A	no-op for managing figure numbers
<code>\figref</code>	A	no-op for managing figure numbers
<code>\figure</code>	PD	figure in current style
<code>\figurebox</code>	PD	stationary boxed figure
<code>\filpage</code>	A	end current page with white space
<code>\floatingfigures</code>	A	figure style is <i>floating</i>
<code>\floatingfigure</code>	PD	define floating figure
<code>\floatingpagefigure</code>	PD	define floating full-page figure
<code>\footdef</code>	A	no-op for managing footnote numbers
<code>\footref</code>	A	no-op for managing footnote numbers
<code>\footnote</code>	SD	specify footnote while in paragraph
<code>\footmatter</code>	PD	specify footnote text with no marker

<code>\ftpar</code>	Par	flush-tagged paragraph
<code>\heading</code>	PD	left-flush heading
<code>\ignore</code>	A	ignore argument
<code>\inline</code>	A	unpadded horizontal line in table
<code>\ipar</code>	Par	indented paragraph
<code>\ivpar</code>	Par	inverted paragraph
<code>\linebreak</code>	A	force line break in paragraph
<code>\midline</code>	A	horizontal line for middle of table
<code>\monthname</code>	A	full name of current month
<code>\monthshortname</code>	A	first three letters of current month
<code>\footnote</code>	SD	specify numbered footnote in paragraph
<code>\oopstex</code>	A	the \TeX symbol
<code>\opentable</code>	PD	unboxed table
<code>\padline</code>	A	put padding in table
<code>\pageeject</code>	A	force a page break in double-column mode
<code>\pagefigure</code>	PD	full page figure in desired style
<code>\partpage</code>	PD	titling page with usual page number
<code>\pbrk</code>	A	escape to vertical mode in LISP programs
<code>\pchapter</code>	PD	chapter title on new page
<code>\quote</code>	PD	quotation text
<code>\regsize</code>	A	switch to the regular size of type
<code>\secdef</code>	A	no-op for managing section numbers
<code>\secref</code>	A	no-op for managing section numbers
<code>\section</code>	PD	section title
<code>\sectionfigures</code>	A	figure style is <i>section</i>
<code>\sectionfigure</code>	PD	define section figure
<code>\sectionpagefigure</code>	PD	define section full-page figure
<code>\setnormalbaselines</code>	A	use <code>\baselinefactor</code> to set baselines
<code>\shortdate</code>	A	date as in 9 Jun 86
<code>\shortyear</code>	A	last two digits of current year
<code>\singlespace</code>	A	single space text
<code>\slashdate</code>	A	date as in 6/9/86
<code>\smlsize</code>	A	switch to a smaller size of type
<code>\smllsize</code>	A	switch to an even smaller size
<code>\smlllsize</code>	A	switch to an even smaller size still
<code>\spread</code>	SD	spread args evenly on line
<code>\stationaryfigures</code>	A	figure style is <i>stationary</i>
<code>\stationaryfigure</code>	PD	define stationary figure
<code>\stationarypagefigure</code>	PD	define stationary full-page figure
<code>\subsection</code>	PD	synonym for <code>\section</code>

<code>\subsubsection</code>	PD	synonym for <code>\section</code>
<code>\sy</code>	SD	set arg in smaller typewriter font
<code>\sybox</code>	SD	hboxed <code>\sy</code>
<code>\tabdef</code>	A	no-op for managing table numbers
<code>\table</code>	PD	vboxed table
<code>\tabref</code>	A	no-op for managing table numbers
<code>\tcenter</code>	PD	centered title
<code>\title</code>	PD	title
<code>\titlepage</code>	PD	titling page with page number 0
<code>\tleft</code>	PD	left-flush title
<code>\topfigures</code>	A	figure style is <i>top</i>
<code>\topfigure</code>	PD	define top figure
<code>\toppagefigure</code>	PD	define top full-page figure
<code>\topline</code>	A	horizontal line for top of table
<code>\tright</code>	PD	right-flush title
<code>\twocolumntext</code>	PD	two-column text
<code>\ucsy</code>	SD	uppercase <code>\sy</code>
<code>\ucsybox</code>	SD	uppercase <code>\sybox</code>
<code>\unvpar</code>	A	allow indentation on next paragraph
<code>\verb</code>	A	verbatim text
<code>\vfootnote</code>	PD	specify footnote text in vertical mode
<code>\vpar</code>	A	disallow indentation on next paragraph
<code>\vtpar</code>	Par	variably-tagged paragraph
<code>\ytext</code>	A	the \TeX symbol

Parameters

The type given in this table is one of I for (integer), D for (dimen), G for (glue), and T for (toks).

Parameter	Type	Default
<code>\abovecaptionskip</code>	G	<code>\medskipamount</code>
<code>\abovechapterskip</code>	G	18pt, stretch is 3 times the <code>\parskip</code> 's
<code>\abstractfont</code>	T	<code>{\smallsize\rm}</code>
<code>\abstractindent</code>	D	0pt
<code>\authorfont</code>	T	<code>{\regsize\rm}</code>
<code>\baselinefactor</code>	T	<code>{1.3}</code>
<code>\belowchapterskip</code>	G	9pt, stretch is 3 times the <code>\parskip</code> 's
<code>\bottomgloss</code>	T	(empty)
<code>\bottomtextfont</code>	T	<code>{\smallsize\rm}</code>
<code>\botsep</code>	T	<code>{\vskip\bigskipamount \footnoterule}</code>
<code>\captionindent</code>	D	<code>\parindent</code>

<code>\captionfont</code>	T	<code>{\smlsize\rm}</code>
<code>\chapterfont</code>	T	<code>{\bigsize\bf}</code>
<code>\columnskip</code>	D	2pc
<code>\copyrightholder</code>	T	<code>{by the author}</code>
<code>\dhbarheight</code>	D	1pt
<code>\dvbarwidth</code>	D	1pt
<code>\figurelinedrop</code>	D	4pt
<code>\footerdrop</code>	D	1.25pc
<code>\footnotefont</code>	T	<code>{\smlsize\rm}</code>
<code>\footnotemarkerwidth</code>	D	10pt
<code>\hbarheight</code>	D	0.4pt
<code>\headerdrop</code>	D	1.25pc
<code>\hour</code>	I	starting hour of job
<code>\intabskip</code>	G	1em
<code>\lispfont</code>	T	<code>{\smlsize\tt}</code>
<code>\minute</code>	I	starting minute of job
<code>\pageno</code>	I	current page number
<code>\partagsep</code>	D	separation between tag and crown line
<code>\postchapterpenalty</code>	I	10000
<code>\posttabskip</code>	G	Opt plus 1fil
<code>\prechapterpenalty</code>	I	-500
<code>\pretabskip</code>	G	Opt plus 1fil
<code>\quotefont</code>	T	<code>{\smlsize\rm}</code>
<code>\rectolefthead</code>	T	<code>{\firstmark}</code>
<code>\rectocenterheader</code>	T	<code><empty></code>
<code>\rectorighthheader</code>	T	<code>{\bf folio}</code>
<code>\rectoleftfooter</code>	T	<code><empty></code>
<code>\rectocenterfooter</code>	T	<code>{\bf folio}</code>
<code>\rectorightfooter</code>	T	<code><empty></code>
<code>\runnerfont</code>	T	<code>{\smlsize\rm}</code>
<code>\runninghead</code>	T	<code><synonym for \rectolefthead></code>
<code>\tablewidth</code>	D	Opt (settable only)
<code>\tablestyle</code>	T	<code>\displaystyle</code>
<code>\titlefont</code>	T	<code>{\bigsize\bf}</code>
<code>\topgloss</code>	T	<code><empty></code>
<code>\topsep</code>	T	<code><empty></code>
<code>\typesize</code>	D	10pt (settable only)
<code>\vbarwidth</code>	D	0.4pt
<code>\versolefthead</code>	T	<code>{\bf folio}</code>
<code>\versocenterheader</code>	T	<code><empty></code>

<code>\versorighthead</code>	T	<code>\langle empty \rangle</code>
<code>\versoleftfoot</code>	T	<code>\langle empty \rangle</code>
<code>\versocenterfoot</code>	T	<code>\{\bf folio\}</code>
<code>\versorightfoot</code>	T	<code>\langle empty \rangle</code>

Switches

The type given in this table is one of **O** for on/off and **T** for true/false.

Switch	Type	Default
<code>\centerheadings</code>	T	false
<code>\figureline</code>	T	false
<code>\footers</code>	O	off
<code>\headers</code>	O	on
<code>\indent</code>	O	on
<code>\runners</code>	O	on
<code>\showcopyright</code>	T	false
<code>\twosided</code>	T	false
<code>\vpar</code>	T	true