

Indian T_EX Users Group

URL: <http://www.river-valley.com/tug>



On-line Tutorial on L^AT_EX

The Tutorial Team

Indian T_EX Users Group, sJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
2000

Prof. (Dr.) K. S. S. Nambooripad, Director, Center for Mathematical Sciences, Trivandrum, (Editor); Dr. E. Krishnan, Reader in Mathematics, University College, Trivandrum; Mohit Agarwal, Department of Aerospace Engineering, Indian Institute of Science, Bangalore; T. Rishi, Focal Image (India) Pvt. Ltd., Trivandrum; L. A. Ajith, Focal Image (India) Pvt. Ltd., Trivandrum; A. M. Shan, Focal Image (India) Pvt. Ltd., Trivandrum; C. V. Radhakrishnan, River Valley Technologies, Software Technology Park, Trivandrum constitute the Tutorial team

This document is generated from L^AT_EX sources compiled with pdfL^AT_EX v. 14e in an INTEL Pentium III 700 MHz system running Linux kernel version 2.2.14-12. The packages used are hyperref.sty and pdfscreen.sty

©2000, Indian T_EX Users Group. This document may be distributed under the terms of the L^AT_EX Project Public License, as described in lplpl.txt in the base L^AT_EX distribution, either version 1.0 or, at your option, any later version



16 Table of Contents, Index and Glossary

16.1 Table of Contents

A *table of contents* is a special list which contains the section numbers and corresponding headings as given in the standard form of the sectioning commands, together with the page numbers on which they begin. Similar lists exist containing reference information about the floating elements in a document, namely, the *list of tables* and *list of figures*. The structure of these lists is simpler, since their contents, the captions of the floating elements, are all on the same level.

Standard L^AT_EX can automatically create these three contents lists. By default, L^AT_EX enters text generated by one of the arguments of the sectioning commands into the `.toc` file. Similarly, L^AT_EX maintains two more files, one for the list of figures (`.lof`) and one for the list of tables (`.lot`), which contain the text specified as the argument of the `\caption` command for figures and tables.

`\tableofcontents` produces a table of contents. `\listoffigures` and `\listoftables` produce a list of figures and list of tables respectively. These lists are printed at the point where these commands are issued. Occasionally, you may find that you don't like the way L^AT_EX prints a table of contents or a list of figures or tables. You can fine-tune an individual entry by using the optional arguments to the sectioning command or `\caption` command that generates it. Formatting commands can also be introduced with the `\addtocontents`. If all else fails, you can edit the `.toc`, `lof`, `lot` files yourself. Edit these files only when preparing the final version of your document, and use a `\nofiles` command to suppress the writing of new versions of the files.

16.1.1 Additional entries

The *-form sectioning commands are not entered automatically in the table of contents. L^AT_EX offers two commands to insert such information directly into a contents file:

```
\addtocontents{file}{text}    \addcontentsline{file}{type}{text}
```

file The extension of the contents file, usually `toc`, `lof` or `lot`.

type The type of the entry. For the `toc` file the *type* is normally the same as the heading according to whose format an entry must be typeset. For the `lof` or `lot` files, `figure` or `table` is specified.

text The actual information to be written to the *file* mentioned. L^AT_EX commands should be protected by `\protect` to delay expansion

The `\addtocontents` command does not contain a *type* parameter and is intended to enter *user-specific* formatting information. For example, if you want to generate additional spacing in the middle of a table of contents, the following command can be issued:

```
\addtocontents{toc}{\protect\vspace{2ex}}
```

The `\addcontentsline` instruction is usually invoked *automatically* by the document sectioning commands, or by the `\caption` commands. If the entry contains numbered text, then `\numberline` must be used to separate the section number (*number*) from the rest of the text for the entry (*heading*) in the *text* parameter:

```
\protect\numberline{number}heading
```

```
\documentclass{article}

\def\bibTeX{\textsc{bib}\TeX}

\begin{document}

\title{\LaTeX{} Guide}
\author{TUG India}
\date{}

\maketitle

\tableofcontents

\addtocontents{toc}{\protect\rule{\textwidth}{.2pt}\par}
\section{Moving Information Around}
\verb+\tableofcontents+ command produces table of contents.....

\section{Bibliography and Citation}
A citation is a cross-reference to another publication, such.....

\subsection{Using \bibTeX}
\bibTeX is a separate program that produces the source list .....

\subsection{Doing it yourself}
A source list is created with the thebibliography .....

\addcontentsline{toc}{section}{\numberline{}}Splitting Your \emph{Input}}
\section*{Splitting Your Input}
\addtocontents{toc}{\noindent\protect\rule{\textwidth}{.2pt}\par}

A large document requires a lot of input. Rather than .....

\end{document}
```

Figure 16.1: Input file contains `\tableofcontents` command.

```
\rule {\textwidth }{.2pt}\par
\contentsline {section}{\numberline
  {1}Moving Information Around}{1}
\contentsline {section}{\numberline
  {2}Bibliography and Citation}{1}
\contentsline {subsection}{\numberline
  {2.1}Using \textsc {bib}\TeX}{1}
\contentsline {subsection}{\numberline
  {2.2}Doing it yourself}{1}
\contentsline {section}{\hbox
  to\@tempdima {\hfil }
  Splitting Your \emph {Input}}{1}
\noindent \rule {\textwidth }{.2pt}\par
```

L ^A T _E X Guide	
TUG India	
Contents	
1 Moving Information Around	1
2 Bibliography and Citation	1
2.1 Using BIB _T E _X	1
2.2 Doing it yourself	1
Splitting Your Input	1
<hr/>	
1 Moving Information Around	
\tableofcontents command produces table of contents. Mainly it does	
2 Bibliography and Citation	
A citation is a cross-reference to another publication, such as a	
2.1 Using bib _T E _X	
bib _T E _X is a separate program that produces the source list for a	
2.2 Doing it yourself	
A source list is created with the thebibliography environment, which	
Splitting Your Input	
A large document requires a lot of input. Rather than putting the	

Figure 16.2: Output `.toc` file and `.dvi`

For example, a `\caption` command inside a `figure` environment saves the text annotating the figure as follows:

```
\addcontentsline{lof}{figure}{\protect\numberline{\thefigure}captioned text}
```

Sometimes `\addcontentsline` is used in the source to complement the actions of standard \LaTeX . For instance, in the case of the starred form of the section commands, no information is written to the `.toc` file. So if you do not want a heading number (starred form) but an entry in the `.toc` file you can write something like:

```
\chapter*{Forward}
\addcontentsline{toc}{chapter}{\numberline{}Forward}
```

This produces an indented “chapter” entry in the table of contents, leaving the space where the chapter number would go free. Omitting the `\numberline` command would typeset the word “Forward” flush left instead.

16.1.2 Typesetting a Contents List

As discussed above, contents lists consists of entries of different types, corresponding to the structural units that they represent. Apart from these standard entries, these lists may contain any commands. A standard entry is specified by the command:

```
\contentsline{type}{text}{page}
```

type type of the entry, e.g. `section`, or `figure`.

text actual text as specified in the argument of the sectioning or `\caption` commands.

page pagenumber.

Note that section numbers are entered as a parameter of the `\numberline` command to allow formatting with the proper indentation. It is also possible for the user to create a table of contents by hand with the help of the command `\contentsline`. For example:

```
\contentsline {section}
{\numberline {2.4}Structure of the Table of Contents}{31}
```

To format an entry in the table of contents files, standard \LaTeX makes use of the following command:

```
\@dottedtocline{level}{indent}{numwidth}{text}{page}
```

The last two parameters coincide with those of `\contentsline`, since the latter usually invokes `\@dottedtocline` command. The other parameters are the following:

level The nesting level of an entry. This parameter allows the user to control how many nesting levels will be displayed. Levels greater than the value of counter `tocdepth` will not appear in the table of contents.

indent This is total indentation from the left margin.

numwidth The width of the box that contains the number if *text* has a `\numberline` command. This is also the amount of extra indentation added to the second and later lines of a multiple line entry.

Additionally, the command `\@dottedtocline` uses the following formatting parameters, which specify the visual appearance of all entries:

- `\@pnumwidth` The width of the box in which the page number is set.
- `\@tocmarg` The indentation of the right margin for all but the last line of multiple line entries. Dimension, but changed with `\renewcommand`.
- `\@dotsep` The separation between dots, in mu (math units). It is a pure number (like 1.7 or 2). By making this number large enough you can get rid of the dots altogether. Changed with `\renewcommand` as well.

16.1.3 Multiple Tables of Contents

The `minitoc` package, initially written by Nigel Ward and Dan Jurafsky and completely redesigned by Jean-Pierre Drucbert, creates a mini-table of contents (a “minitoc”) at the beginning of each chapter when you use the `book` or `report` classes.

The mini-table of contents will appear at the beginning of a chapter, after the `\chapter` command. The parameters that govern the use of this package are discussed below:

Table 16.1: Summary of the `minitoc` parameters

<code>\dominitoc</code>	must be put just in front of <code>\tableofcontents</code> , to initialize the <code>minitoc</code> system (Mandatory).
<code>\faketableofcontents</code>	this command replaces <code>\tableofcontents</code> when you want <code>minitocs</code> but not table of contents.
<code>\minitoc</code>	this command must be put right after each <code>\chapter</code> command where a <code>minitoc</code> is desired.
<code>minitocdepth</code>	a \LaTeX counter that indicates how many levels of headings will be displayed in the <code>minitoc</code> (default value is 2).
<code>\mtcindent</code>	the length of the left/right indentation of the <code>minitoc</code> (default value is 24pt).
<code>\mtcfont</code>	command defining the font that is used for the <code>minitoc</code> entries (The default definition is a small roman font).

For each mini-table, an auxiliary file with extension `.mtc<N>` where `<N>` is the chapter number, will be created.

By default, these mini-tables contain only references to sections and subsections. The `minitocdepth` counter, similar to `tocdepth`, allows the user to modify this behaviour.

As the `minitoc` takes up room on the first page(s) of a chapter, it will alter the page numbering. Therefore, three runs normally are needed to get correct information in the mini-table of contents.

To turn off the `\minitoc` commands, merely replace the package `minitoc` with `minitocoff` on your `\usepackage` command. This assures that all `\minitoc` commands will be ignored.

16.2 Index

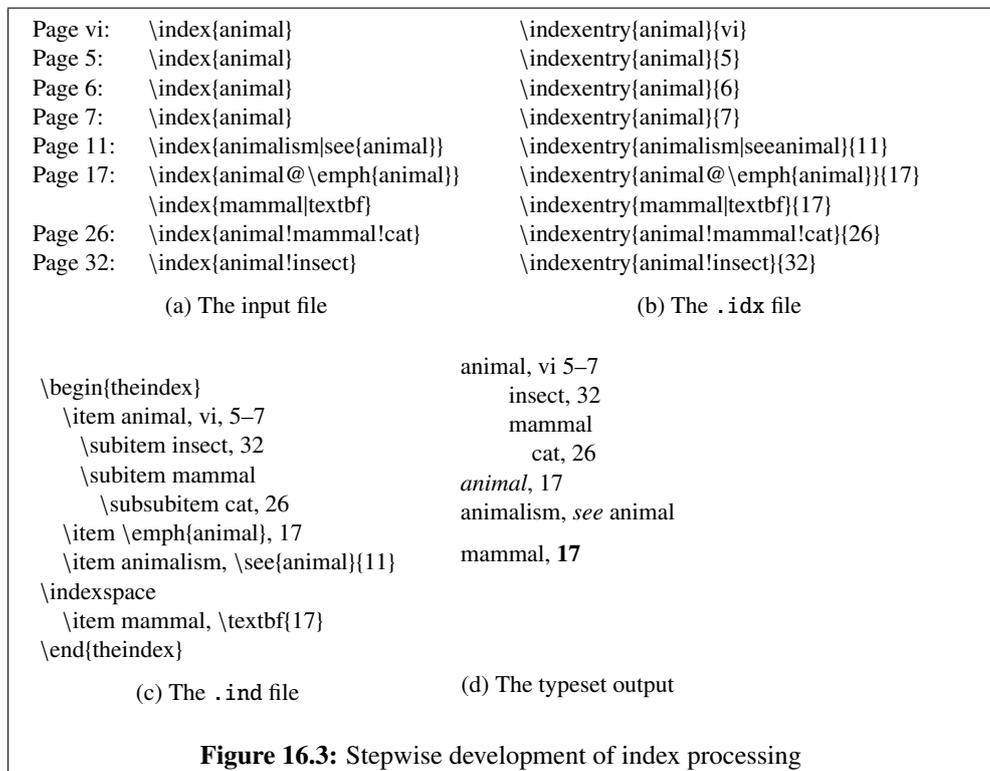
To find a topic of interest in a large document, book, or reference work, you usually turn to the table of contents or, more often, to the index. Therefore, an index is a very important part of a document, and most users' entry point to a source of information is precisely through a pointer in the index. The most generally used index preparation program is *MakeIndex*.

Each `\index` command causes \LaTeX to write an entry in the `.idx` file. This command writes the text given as an argument, in the `.idx` file. This `.idx` will be generated only if we give `\makeindex` command in the preamble otherwise it will produce nothing.

```
\index{index_entry}
```

To generate Index follow the procedure given below:

- (1) Tag the words inside the document, which needs to come as index, as an argument of `\index` command.
- (2) Include the `makeidx` package with an `\usepackage` command and put `\makeindex` command at the preamble.
- (3) Put a `\printindex` command where the index is to appear, normally before `\end{document}` command.
- (4) \LaTeX *file*. Then a raw index (`file.idx`) will be generated.
- (5) Then Run `makeindex`. (`makeindex file.idx` or `makeindex file`). Then two more files will be generated, `file.ind` which contains the index entries and `file.ilg`, a transcript file.
- (6) Then again run \LaTeX . Now you can see in the dvi that the Index has been generated in a new page.



16.2.1 Simple Index Entries

Each `\index` command causes \LaTeX to write an entry in the `.idx` file. For example

```
\index{index_entry}
```

16.2.2 Sub Entries

Up to three levels of index entries (main, sub, and subsub entries) are available with \LaTeX -MakeIndex. To produce such entries, the argument of the `\index` command should contain both the main and subentries, separated by `!` character.

Page 5: `\index{dimensions!rule!width}`

This will come as

```
dimensions
  rule
  width, 5
```

16.2.3 Page Ranges and Cross-References

You can specify a page range by putting the command `\index{...|{}` at the beginning of the range and `\index{...|)}` at the end of the range. Page ranges should span a homogeneous numbering scheme (e.g., roman and arabic page numbers cannot fall within the same range).

You can also generate cross-reference index entries without page numbers by using the see encapsulator. Since “see” entry does not print any page number, the commands `\index{...|see{...}}` can be placed anywhere in the input file after the `\begin{document}` command. For practical reasons, it is convenient to group all such cross-referencing commands in one place.

fonts	Page ii:	<code>\index{table {}</code>
Computer Modern, 13–25	Page xi:	<code>\index{table)}</code>
math, <i>see</i> math, fonts	Page 5:	<code>\index{fonts!PostScript {}</code>
PostScript, 5		<code>\index{fonts!PostScript)}</code>
table, ii–xi, 14	Page 13:	<code>\index{fonts!Computer Modern {}</code>
	Page 14:	<code>\index{table }</code>
	Page 17:	<code>\index{fonts!math see{math, fonts}}</code>
	Page 21:	<code>\index{fonts!Computer Modern }</code>
	Page 25:	<code>\index{fonts!Computer Modern)}</code>

Figure 16.4: Page range and cross-referencing

16.2.4 Controlling the Presentation Form

Sometimes you may want to sort an entry according to a key, while using a different visual representation for the typesetting, such as Greek letters, mathematical symbols, or specific typographic forms. This function is available with the syntax: `key@visual`, where *key* determines the alphabetical position and the string *value* produces the typeset text of the entry.

For some, indexes, certain page numbers should be formatted specially, with an italic page number (for example) indicating a primary reference, and an *n* after a page number denoting that the item appears in a footnote on that page. *MakeIndex* allows you to format an individual page number in any way you want by using the encapsulator syntax specified | character. What follows the | sign will “encapsulate” or enclose the page number associated with the index entry. For instance, the command `\index{keyword|xxx}` will produce a page number of the form `\xxx{n}`, where *n* is the page number in question. Similarly, the command `\index{keyword|(xxx)}` will generate a page range of the form `\xxx{n-m}`

`\newcommand{\nn}[1]{#1n}`

delta, 14	Page ii:	<code>\index{tabular textbf}</code>
δ , 23	Page 5:	<code>\index{ninety-five}</code>
delta wing, 16	Page 7:	<code>\index{tabbing}</code>
flower , 19	Page 14:	<code>\index{delta}</code>
ninety, 26	Page 16:	<code>\index{delta wing}</code>
xc, 28	Page 19:	<code>\index{flower@ textbf{flower}}</code>
ninety-five, 5	Page 21:	<code>\index{tabular textit}</code>
tabbing, 7, 34–37	Page 22:	<code>\index{tabular nn}</code>
tabular, ii, 21, 22n	Page 23:	<code>\index{delta@δ}</code>
tabular environment, 23		<code>\index{tabular@ texttt{tabular} environment}</code>
	Page 26:	<code>\index{ninety}</code>
	Page 28:	<code>\index{ninety@xc}</code>
	Page 34:	<code>\index{tabbing (textit)}</code>
	Page 36:	<code>\index{tabbing })</code>

Figure 16.5: Controlling the presentation form

16.2.5 Printing those Special Characters

To typeset one of the characters having a special meaning to *MakeIndex* (!, ", @, or |) in the index, precede it with a " character. More precisely, any character is said to be quoted if it follows an unquoted " that is not part of a \ command. The latter case is for allowing umlaut characters. Quoted !, @, ", or | characters are treated like ordinary characters, losing their special meaning. The " preceding a quoted character is deleted before the entries are alphabetised.

@ sign, 2		<code>\index{bar@ texttt{"}} see{vertical bar}}</code>
, <i>see</i> vertical bar	Page 1:	<code>\index{quote (\verb+""+)}</code>
exclamation (!), 4		<code>\index{quote@ texttt{""} sign}</code>
Ah!, 5	Page 2:	<code>\index{atsign@ texttt{"@} sign}</code>
Mädchen, 3	Page 3:	<code>\index{maedchen@M""{a}dchen}</code>
quote ("), 1	Page 4:	<code>\index{exclamation ("!)}</code>
" sign, 1	Page 5:	<code>\index{exclamation ("!)!Ah"!}</code>

16.6 Printing those special characters

16.3 Glossary

A 'glossary' is a special index of terms and phrases alphabetically ordered together with their explanations. To help set up a glossary, L^AT_EX offers the commands

<code>\makeglossary</code>	in the preamble and
<code>\glossary{glossary-entry}</code>	in the text part

which function just like the commands for making up an index register. The entries are written to a file with extension `.glo` after the command `\makeglossary` has been given in the preamble. The form of these file entries from each `\glossary` command is

```
\glossaryentry{glossary-entry}{pagenumber}
```

The information the `.glo` file can be used to establish a glossary. However, there is no equivalent to the `theindex` environment for a glossary, but a recommended structure is the `description` environment or a special list environment.