

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; 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 lpp1.txt in the base L^AT_EX distribution,
either version 1.0 or, at your option, any later version



6 Floats

6.1 Table

With the *box* elements already explained in the previous chapter, it would be possible to produce all sorts of framed and unframed tables. However, \LaTeX offers the user far more convenient ways to build such complicated structures.

6.1.1 Constructing tables

The environments `tabular` and `tabular*` are the basic tools with which tables can be constructed. The syntax for these environments is:

```
\begin{tabular}[pos]{cols} rows \end{tabular}
\begin{tabular*}{width}[pos]{cols} rows \end{tabular*}
```

Both the above environments actually create a minipage. The meaning of the above arguments is as follows:

pos Vertical positioning arguments (see also the explanation of this argument for `parboxes`). It can take on the values.

- t** the top line of the table is aligned with the baseline of the current external line of text
- b** the bottom line of the table is aligned with the external baseline

with no positioning argument given, the table is centered on the external baseline.

width This argument applies to only the `tabular*` environment and determines its overall width. In this case, the `cols` argument must contain the @-expression (see below) `@{\extracolsep{\fill}}` somewhere after the first entry. For the other two environments, the total width is fixed by the textual content.

cols The column formatting argument. There must be an entry for every column, as well as possible extra entries for the left and right borders of the table or for the inter-column spacings.

The possible *column formatting symbols* are:

- l** the column contents are left justified
- c** the column contents are centered
- r** the column contents are right justified
- p{wd}** the text in this column is set into lines of width *wd* and the top line is aligned with the other columns. In fact, the text is set in a parbox with the command `\parbox[t]{wd}{column text}`
- *{num}{cols}** the column format contained in *cols* is reproduced *num* times, so that `*{3}{|c|}` is the same as `|c|c|c|`.

The available formatting symbols for right and left borders and for the inter-column spacing are:

- | draws a vertical line
- || draws two vertical lines next to each other

@{text} this entry is referred to as an @-expression, and inserts text in every line of the table between the two columns where it appears.

@-expression removes the inter-column spacing that is automatically put between each pair of columns. If white space is needed between the inserted text and the next column, this must be explicitly included with `\hspace{ }` within the text of the @-expression. If the inter-column spacing between two particular columns is to be something other than the standard, this may be easily achieved by placing `@{\hspace{wd}}` between the appropriate columns in the formatting argument. This replaces the standard inter-column spacing with the width *wd*.

An `\extracolsep{wd}` within an @-expression will put extra spacing of amount *wd* between all the following columns, until countermanded by another `\extracolsep` command. In contrast to the standard spacing, this additional spacing is not removed by later @-expression. In the `\tabular*` environment, there must be a command `@{\extracolsep\fill}` somewhere in the column format so that all the subsequent inter-column spacings can stretch out to fill the predefined table width.

If the left or right borders of the table do not consist of a vertical line, spacing is added there of an amount equal to half the normal inter-column spacing. If this spacing is not required, it may be suppressed by including an empty @-expression `@{}` at the beginning or end of the column format.

rows contain the actual entries in the table, each horizontal row being terminated with `\`. These rows consist of a sequence of column entries separated from each other by the `&` symbol. Thus each row in the table contains the same number of column entries as in the column definition *cols*. Some entries may be empty. The individual column entries are treated by \LaTeX as though they were enclosed in braces `{ }`, so that any changes in type style or size are restricted to that one column.

\hline This command may only appear before the first row or immediately after a row termination `\`. It draws a horizontal line the full width of the table below the row that was just ended, or at the top of the table if it comes at the beginning. Two `\hline` commands together draw two horizontal lines with a little space between them.

\cline{n-m} This command draws a horizontal line from the left side of column *n* to the right side of column *m*. Like `\hline`, it may only be given just after a row termination `\`, and there may be more than one after another. The command `\cline{1-3}` `\cline{5-7}` draws two horizontal lines from column 1 to 3 and from column 5 to 7, below the row that was just ended. In each case, the full column widths are underlined.

\vline This command draws a vertical line with the height of the row at the location where it appears. In this way, vertical lines that do not extend the whole height of the table may be inserted with a column.

\multicolumn{num}{col}{text} This command combines the following *num* columns into a single column with their total width including inter-column spacing. The argument *col* contains exactly one of the positioning symbols `l`, `r`, `c`, with possible @-expressions and vertical lines `|`. A value of 1 may be given for *num* when the positioning argument is to be changed for that column in one particular row.

In this context, a ‘column’ starts with a positioning symbol `l`, `r`, or `c` and includes everything upto but excluding the next one. The first column also includes everything before the first positioning symbol. Thus `c@{}r|` contains three columns: the first is `|c@{}`, the second `r`, and the third `r|`.

6.2 Table style parameters

There are a number of style parameters used in generating tables which \LaTeX sets to standard values. These may be altered by the user, either globally within the preamble or locally inside an environment. They should not be changed within the `tabular` environment.

- `\tabcolsep` is half the width of the spacing that is inserted between columns in the `tabular` and `tabular*` environments.
- `\arrayrulewidth` is the thickness of the vertical and horizontal lines within a table.
- `\doublerulesep` is the separation between the lines of a double rule.
- `\arraystretch` can be used to change the distance between the rows of a table. This is a multiplying factor, with a standard value of 1. A value of 1.5 means that the inter-row spacing is increased by 50%. A new value is set by redefining the parameters with the command:

```
\renewcommand{\arraystretch}{\factor}
```

Following are the commands for changing the table style parameters that relate to dimensions:

```
\setlength\tabcolsep{\dimen}
\setlength\arrayrulewidth{\dimen}
\setlength\doublerulesep{\dimen}
```

6.3 Example

Creating tables is much easier in practice than it would seem from the above list of formatting possibilities. This is best illustrated with an example.

The simplest table consists of a row of columns in which the text entries are either centered or justified to one side. The column widths, the spacing between the columns, and thus the entire width of the table are automatically calculated.

Sample Tabular		
col head	col head	col head
Left	centered	right
aligned	items	aligned
items	items	items
Left items	centered	right aligned

See the code that generated the table above.

```
\begin{tabular}{|l|c|r|}
\hline
\multicolumn{3}{|c|}{Sample Tabular}\hline
col head & col head & col head\hline
\hline Left & centered & right \\ \cline{1-2}
aligned & items & aligned \\ \cline{2-3}
items & items & items \\ \cline{1-2}
Left items & centered & right aligned \\
\hline
\end{tabular}
```

The discussion on tables doesn't conclude with this chapter, instead more bells and whistles

are to be discussed, that includes long tables (table that spans multiple pages), how to repeat the column headings and special footlines in all multipaged tables, color tables and also few other embellishments, which the scientific community at large might require in their document preparation. So watch out for the next chapter.

6.4 Exercise

Here is an exercise you can try. If you encounter any problems, please get back to the list.

Plan for \TeX Users Group 2001–2003						
Project	No. <input type="text"/> <input type="text"/> <input type="text"/>		Name <input type="text"/>			
Year	2001		2002		2003	
	Rs.	US\$	Rs.	US\$	Rs.	US\$
Internet costs						
Journal costs						
\TeX Live production costs						
Signature				Authorization		