The longtable package*

David Carlisle

1995/06/15

Abstract

This package defines the `longtable` environment, a multi-page version of `tabular`.

List of Tables

1 Introduction

`longtable` `longtable.sty` defines a new environment, `longtable`, which has most of the features of the `tabular` environment, but produces tables which may be broken by TeX's standard page-breaking algorithm. It also shares some features with the `table` environment. In particular it uses the same counter, `table`, and has a similar `\caption` command. Also, the standard `\listoftables` command lists tables produced by either the `table` or `longtable` environments.

The following example uses most of the features of the `longtable` environment. An edited listing of the input for this example appears in Section 8.

Note: Various parts of the following table will **not** line up correctly. This is to demonstrate a feature of this package. Tables are not aligned unless a `\setlongtables` declaration is used (usually in the preamble) or equivalently on of the package options `set` or `final` is specified. See page 4 for more details.

Table 1: A long table

This part appears at the top of the table			
* First	Second *		
*`longtable` columns are specified	in the	*	
* same way as in the `tabular`	environment.	*	
* `@{*}r||p{1in}@{*}`	in this case.	*	
* Each row ends with a	`\\` command.	*	
* The `\\` command has an	optional	*	
*This goes at the	bottom. *		

*This file has version number v3.15, last revised 1995/06/15.

Table 1: (continued)

This part appears at the top of every other page	
* **First**	**Second** *
* argument, just as in	the *
* tabular	environment. *
* See the effect of \\[10pt]	? *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Also \hline may be used,	as in tabular. *
* That was a \hline	. *
* That was \hline\hline	. *
This is a \multicolumn{2}{||c||}	
*If a page break occurs at a \hline then	a line is drawn *
* at the bottom of one page and at the	top of the next. *
*The [t] [b] [c] argument of tabular	can not be used.*
* The optional argument may be one of	[l] [r] [c] *
* to specify whether the table should be	adjusted *
* to the left, right	or centrally. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
* Lots of lines	like this. *
*This goes at the	bottom. *

Table 1: (continued)

This part appears at the top of every other page			
*	**First**	**Second**	*
*Some lines may take up a lot of space, like this:		This last column is a "p" column so this "row" of the table can take up several lines. Note however that TEX will never break a page within such a row. Page breaks only occur between rows of the table or at `\hline` commands.	*
*	Lots of lines	like this.	*
*	Lots of lines	like this.	*
*	Lots of lines	like this.	*
*	Lots of lines	like this.	*
*	Lots of lines	like this.	*
*	Lots of lines	like this.	*
*	Lots of lines	like this.	*
*	Lots[1] of lines	like this.	*
*	Lots of lines	like this[2]	*
*	Lots of lines	like this.	*
*	Lots of lines	like this.	*
*	These lines will	appear	*
*	in place of the	usual foot	*
*	at the end	of the table	*

2 Chunk Size

`\LTchunksize` In order to TEX multi-page tables, it is necessary to break up the table into smaller chunks, so that TEX does not have to keep everything in memory at one time. By default `longtable` uses 20 rows per chunk, but this can be set by the user, with e.g., `\LTchunksize=10`.[3] These chunks do not affect page breaking, thus if you are

[1] This is a footnote.

[2] `longtable` takes special precautions, so that footnotes may also be used in 'p' columns.

[3] Since V3.07, you can also use the LATEX syntax `\setcounter{LTchunksize}{10}`.

A	`tabular`	environment
within	a floating	`table`

Table 2: A floating table

using a TeX with a lot of memory, you can set `\LTchunksize` to be several pages of the table. TeX will run faster with a large `\LTchunksize`. However, if necessary, `longtable` can work with `\LTchunksize=1`, in which case the memory taken up is negligible. Note that if you use the commands for setting the table head or foot (see below), the `\LTchunksize` must be at least as large as the number of rows in each of the head or foot sections.

This document specifies `\LTchunksize=10`. If you look at the previous table, you will see that various parts of the table do not line up. Also LaTeX will have printed a warning that the column widths had changed. `longtable` writes information onto the `.aux` file, so that it can line up the different chunks; however,

`\setlongtables`

it does not use this information unless a `\setlongtables` command is given before the table starts. It is best not to use this command until the document is nearly complete, as otherwise `longtable` will leave enough room in each column for the widest entry *in all previous versions*. (At least since the `.aux` file was last removed.)

If you look at the preamble to this document, you will see a line:
`%\setlongtables`
If you delete the `%` and re-LaTeX the document, all the tables will line up perfectly. (You may need to LaTeX the document twice if the `.aux` file has been destroyed.)

3 Captions and Headings

`\endhead` `\endfirsthead` `\endfoot` `\endlastfoot`

At the start of the table one may specify lines which are to appear at the top of every page (under the headline, but before the other lines of the table). The lines are entered as normal, but the last `\\` command is replaced by a `\endhead` command. If the first page should have a different heading, then this should be entered in the same way, and terminated with the `\endfirsthead` command. The `\LTchunksize` should be at least as large as the number of rows in the heading. There are also `\endfoot` and `\endlastfoot` commands which are used in the same way (at the *start* of the table) to specify rows (or an `\hline`) to appear at the bottom of each page. In certain situations, you may want to place lines which logically belong in the table body at the end of the `firsthead`, or the beginning of the `lastfoot`. This helps to control which lines appear on the first and last page of the table.

`\caption`

The `\caption{...}` command is essentially equivalent to
`\multicolumn{n}{c}{\parbox{\LTcapwidth}{...}}`
where `n` is the number of columns of the table. You may set the width of the caption with a command such as `\setlength{\LTcapwidth}{3in}` in the preamble of your document. The default is 4in. `\caption` also writes the information to produce an entry in the list of tables. As with the `\caption` command in the `figure` and `table` environments, an optional argument specifies the text to appear in the list of tables if this is different from the text to appear in the caption. Thus the caption for table 1 was specified as `\caption[An optional caption (for the`

`list of tables)]{A long table\label{long}}`.

You may wish the caption on later pages to be different to that on the first page. In this case put the `\caption` command in the first heading, and put a subsidiary caption in a `\caption[]` command in the main heading. If the optional argument to `\caption` is empty, no entry is made in the list of tables. Alternatively, if you do not want the table number to be printed each time, use the `\caption*` command.

The captions are set based on the code for the article class. If you have redefined the standard `\@makecaption` command to produce a different format for the captions, you may need to make similar changes to the longtable version, `\LT@makecaption`. See the code section for more details.

You may use the `\label` command so that you can cross reference `longtable`s with `\ref`. Note however, that the `\label` command should not be used in a heading that may appear more than once. Place it either in the `firsthead`, or in the body of the table. It should not be the *first* command in any entry.

4 Multicolumn entries

`\multicolumn` entries present a particular problem for `longtable`. Suppose we were trying to typeset the following table, with`\LTchunksize=1`.

Table 3: A good example

	A wide multicolumn entry
A wide column in a different chunk	narrow

If after the first chunk, `longtable` had only seen the wide multicolumn entry, then it would think that the first column was very narrow (just containing the rules and glue specified by `|l|`). All the width of the multicolumn entry would be assumed to be in the second column. (This is a 'feature' of TEX's primitive `\halign` command.) `longtable` would then pass the information that there is a wide second column to the second chunk, with the result that the final table would be much too wide, like this:

Table 4: A bad example

	A wide multicolumn entry
A wide column in a different chunk	narrow

`\kill` The solution to this problem used in Versions 1 and 2, was to use a `\kill` line. If a line is `\kill`ed, by using `\kill` rather than `\\` at the end of the line, it is used in calculating column widths, but removed from the final table. Thus entering a `\kill`ed copy of the second row before the multicolumn entry would mean that `\halign` 'saw' the wide entry in the first column, and so did not widen the second column to make room for the multicolumn entry.

In Version 3, a new solution has been introduced. If the saved row in the `.aux` file is not being used, `longtable` uses a special 'draft' form of `\multicolumn`, this modifies the definition, so the spanning entry is never considered to be wider than the columns it is spanning. So after the first pass, the `.aux` file stores the widest normal entry for each column, no column has been widened due to \spanned columns. By default longtable ignores the .aux file, and so each run of LaTeX is considered a first pass. Once the \setlongtables declaration is given, the saved row in the .aux file, and the proper definition of \multicolumn are used. If any \multicolumn entry causes one of the columns to be widened, this information can not be passed back to earlier chunks, and so the table will not correctly line up until the third pass.

This new system means that \kill is essentially redundant, however I think that some people will object to LaTeX-ing a file three times, and so I have made it easy to revert to the Version 2 behaviour. Giving the declaration \let\LTmulticolumn=\multicolumn will stop multicolumn entries being set in draft mode, and so in order to avoid bad examples like Table 4, you will need to use \kill as discussed above.

If you object even to LaTeX-ing a file twice, you should disable \LTmulticolumn as above, and also make the first line of every longtable a \kill line that contains the widest entry to be used in each column. All chunks will then line up on the first pass, and \setlongtables need not be used at all.

5 Adjustment

\LTleft
\LTright

The last two tables also show the use of the optional argument. They are set with \begin{longtable}[c] ... specifying that the table should be centred. The other possible options are [r] and [l], for right and left adjustment, respectively. Normally centering is the default, but this document specifies \LTleft=\parindent \LTright=\fill in the preamble,[4] which means that the tables are set flush left, but indented by the usual paragraph indentation. Any lengths can be specified for these two parameters, but at least one of them should be a rubber length so that it fills up the width of the page, unless rubber lengths are added between the columns using the \extracolsep command.

6 Changes Since Version 2

There have been several changes made between versions 2 and 3.

- The mechanism for adding the head and foot of the table has been completely rewritten. With this new mechanism, longtable does not need to issue a \clearpage at the start of the table, and so the table may start half way down a page. Also the \endlastfoot command which could not safely be implemented under the old scheme, has been added.

- longtable now issues an error if started in the scope of \twocolumn, or the multicols environment.

- The separate documentation file longtable.tex has been merged with the style file, longtable.sty using Mittelbach's doc.sty.

- Support for footnotes has been added. Note however that \footnote will not work in the 'head' or 'foot' sections of the table. In order to put a footnote

[4] Equivalent to \setlength{\LTleft}{\parindent} \setlength{\LTright}{\fill}

in those sections (e.g., inside a caption), use \footnotemark at that point, and \footnotetext anywhere in the table *body* that will fall on the same page.

- The treatment of \multicolumn has changed, making \kill lines unnecessary, at the price of sometimes requiring a third pass through LaTeX.
- The \newpage command now works inside a longtable.

In particular, the first of these changes means that Version 3 may not be able to exactly reproduce documents originally developed with Version 2, even if \clearpage commands are placed before and after the longtable environment. For this reason, individual users, *not archives*, may want to retain Version 2 under a different name if they have documents using Version 2.

7 Summary

Table 5: A summary of longtable commands

Parameters		
\LTleft	Glue to the left of the table.	(\fill)
\LTright	Glue to the right of the table.	(\fill)
\LTpre	Glue before the the table.	(\bigskipamount)
\LTpost	Glue after the the table.	(\bigskipamount)
\LTchunksize	The number of rows per chunk.	(20)
\LTcapwidth	The width of a parbox containing the caption.	(4in)
Setlongtables		
\setlongtables	Use column widths from the previous run.	
Optional arguments to \begin{longtable}		
none	Position as specified by \LTleft and \LTright.	
[c]	Centre the table.	
[l]	Place the table flush left.	
[r]	Place the table flush right.	
Commands available inside longtable		
\endhead	Specifies rows to appear at the top of every page.	
\endfirsthead	Specifies rows to appear at the top the first page.	
\endfoot	Specifies rows to appear at the bottom of every page.	
\endlastfoot	Specifies rows to appear at the bottom of the last page.	
\kill	Row is 'killed', but is used in calculating widths.	
\caption{foo}	Caption 'Table ?: foo', and a 'foo' entry in the list of tables.	
\caption[bar]{foo}	Caption 'Table ?: foo', and a 'bar' entry in the list of tables.	
\caption[]{foo}	Caption 'Table ?: foo', but no entry in the list of tables.	
\caption*{foo}	Caption 'foo', but no entry in the list of tables.	
\newpage	Force a page break.	
\footnote	Footnotes, but may not be used in the table head & foot.	
\footnotemark	Footnotemark, May be used in the table head & foot.	
\footnotetext	Footnote text, Use in the table body.	

8 Verbatim highlights from Table 1

```
\begin{longtable}{@{*}r||p{1in}@{*}}
KILLED & LINE!!!! \kill
\caption[An optional caption ...]{A long table\label{long}}\\
\hline\hline
\multicolumn{2}{@{*}c@{*}}%
     {This part appears at the top of the table}\\
\sc First&\sc Second\\
\hline\hline
\endfirsthead
\caption[]{(continued)}\\
\hline\hline
\multicolumn{2}{@{*}c@{*}}%
      {This part appears at the top of every other page}\\
\bf First&\bf Second\\
\hline\hline
\endhead
\hline
This goes at the&bottom.\\
\hline
\endfoot
\hline
These lines will&appear\\
in place of the & usual foot\\
at the end& of the table\\
\hline
\endlastfoot
{\tt longtable}  columns  are specified& in the \\
same way as  in the {\tt tabular}& environment.\\
...
\multicolumn{2}{||c||}{This is a ...}\\
...
Lots of lines& like this.\\
...
\hline
Lots\footnote{...} of lines& like this.\\
Lots   of   lines& like this\footnote{...}\\
\hline
Lots of lines& like this.\\
...
\end{longtable}
```

9 The Macros

1 ⟨*package⟩

9.1 Initial code

Before declaring the package options, we must define some defaults here.

`\LT@err` The error generating command

```
\def\LT@err{\PackageError{longtable}}
```

`\LT@err` The warning generating command

```
\def\LT@warn{\PackageWarning{longtable}}
```

9.2 Options

The first two options deal with error handling. They are compatible with the options used by the `tracefnt` package.

`errorshow` *Only* show errors on the terminal. 'warnings' are just sent to the log file.

```
\DeclareOption{errorshow}{%
  \def\LT@warn{\PackageInfo{longtable}}}
```

`pausing` Make every warning message into an error so TeX stops. May be useful for debugging.

```
\DeclareOption{pausing}{%
  \def\LT@warn#1{%
    \PackageError{longtable}{#1}{This is not really an error}}}
```

`set` `final` The next options are just alternative syntax for the `\setlongtables` declaration.

```
\DeclareOption{set}{%
  \AtEndOfPackage{\setlongtables\let\setlongtables\relax}}
\DeclareOption{final}{%
  \ExecuteOptions{set}}
```

```
\ProcessOptions
```

9.3 User Setable Parameters

`\LTleft` `\LTright` Glue to the left and right of the table, default `\fill` (ie centred).

```
\newskip\LTleft        \LTleft=\fill
\newskip\LTright       \LTright=\fill
```

`\LTpre` `\LTpost` Glue before and after the `longtable`. `\bigskip` by default.

```
\newskip\LTpre         \LTpre=\bigskipamount
\newskip\LTpost        \LTpost=\bigskipamount
```

`\LTchunksize` Chunk size (The number of rows taken per `\halign`). Default 20.

```
\newcount\LTchunksize  \LTchunksize=20
```

`\c@LTchunksize` Added in V3.07 to allow the purist syntax `\setcounter{LTchunksize}{10}`.

```
\let\c@LTchunksize\LTchunksize
```

`\LTcapwidth` Width of the `\parbox` containing the caption. Default 4in.

```
\newdimen\LTcapwidth   \LTcapwidth=4in
```

9.4 Internal Parameters

\LT@head \LT@firsthead \LT@foot \LT@lastfoot

Boxes for the table head and foot.

```
\newbox\LT@head
\newbox\LT@firsthead
\newbox\LT@foot
\newbox\LT@lastfoot
```

\LT@cols

Counter for number of columns.

```
\newcount\LT@cols
```

\LT@rows

Counter for rows up to chunksize.

```
\newcount\LT@rows
```

\c@LT@tables

Counter for the tables, added in V3.02. Previous versions just used the LaTeX counter `table`, but this fails if `table` is reset during a document, eg `report` class resets it every chapter.

This was changed from `\newcount\LT@tables` in V3.04. LaTeX counters are preserved correctly when `\includeonly` is used. In the rest of the file `\LT@tables` has been replaced by `\c@LT@tables` without further comment. Don't ask what `\relax` is doing here!

```
\newcounter{LT@tables}\relax
```

\c@table \fnum@table \tablename

If the `table` counter is not defined (eg in `letter` style), define it. (Added in V3.06.)

```
\ifx\c@table\undefined
  \newcounter{table}
  \def\fnum@table{\tablename~\thetable}
\fi
\ifx\tablename\undefined
  \def\tablename{Table}
\fi
```

\LT@out

In a normal style, `longtable` uses the `.aux` file to record the column widths. With `letter.sty`, use a separate `.lta` file. (Added in V3.06.)

Not needed for new letter class.

```
\ifx\startlabels\undefined
  \let\@auxout\@auxout
\else
  {\@input{\jobname.lta}}%
  \newwrite\@auxout
  \immediate\openout\@auxout=\jobname.lta
\fi
```

\LT@p@ftn

Temporary storage for footnote text in a 'p' column.

```
\newtoks\LT@p@ftn
```

\LT@end@pen

Special penalty for the end of the table. Done this way to save using up a count register.

```
\mathchardef\LT@end@pen=30000
```

9.5 The longtable environment

\longtable

Called by \begin{longtable}. This implementation does not work in multiple column formats. \par added at V3.04.

```
\def\longtable{%
  \par
  \ifx\multicols\@undefined\else
     \ifnum\col@number>\@ne\@twocolumntrue\fi\fi
  \if@twocolumn\LT@err{longtable not in 1-column mode}\@ehc\fi
  \begingroup
```

Check for an optional argument.

```
  \@ifnextchar[\LT@array{\LT@array[x]}}
```

\LT@array

Start setting the alignment. Based on \@array from latex.tex and array.sty.

Since Version 3.02, longtable has used the internal counter \c@LT@tables. The LaTeX counter table is still incremented so that \caption works correctly.

```
\def\LT@array[#1]#2{%
  \refstepcounter{table}\global\advance\c@LT@tables\@ne
```

Set up the glue around the table if an optional argument given.

```
  \if l#1\LTleft\z@ \LTright\fill
   \else\if r#1\LTleft\fill \LTright\z@
     \else\if c#1\LTleft\fill \LTright\fill \fi\fi\fi
```

Set up these internal commands for longtable.

```
  \global\let\LT@mcw@rn\relax\let\LT@mcol\multicolumn
  \let\\\LT@tabularcr\let\tabularnewline\\%
  \def\newpage{\noalign{\break}}%
  \let\hline\LT@hline \let\kill\LT@kill\let\caption\LT@caption
  \@tempdima\ht\strutbox
```

Set up internal commands according to Lamport or Mittelbach.

```
  \ifx\extrarowheight\LT@undef
```

Initialise these commands as in latex.tex.

```
    \let\@acol\@tabacol
    \let\@classz\@tabclassz \let\@classiv\@tabclassiv
    \def\@startpbox{\vtop\LT@startpbox}\let\LT@LL@FM@cr\@tabularcr
  \else
```

Initialise these commands as in array.sty. \d@llar replaced by \d@llarbegin \d@llarend in V3.03 to match array.sty V2.0h. We do not need to set \d@llarbegin and \d@llarend as array.sty gives them the correct values at the top level.

```
    \advance\@tempdima\extrarowheight
    \col@sep\tabcolsep
    \let\@startpbox\LT@startpbox\let\LT@LL@FM@cr\@arraycr
  \fi
```

The rest of this macro is mainly based on array.sty, but should work for the standard tabular too.

```
  \setbox\@arstrutbox\hbox{\vrule
    \@height \arraystretch \@tempdima
    \@depth \arraystretch \dp \strutbox
    \@width \z@}%
  \let\@sharp##\let\protect\relax
```

Interpret the preamble argument.

```
  \begingroup
   \@mkpream{#2}%
```

We need to rename `\@preamble` here as F.M.'s scheme uses `\global`, and we may need to nest `\@mkpream`, eg for `\multicolumn` or an `array`. We do not need to worry about nested longtables though!

```
   \xdef\LT@bchunk{\global\LT@rows\z@\setbox\z@\vbox\bgroup
      \tabskip\LTleft\halign to\hsize\bgroup
     \tabskip\z@ \@arstrut \@preamble \tabskip\LTright \cr}%
  \endgroup
```

Find out how many columns we have (store in `\LT@cols`).

```
  \expandafter\LT@nofcols\LT@bchunk&\LT@nofcols
```

Get the saved row from `\LT@i`…`\LT@ix` (from the `.aux` file), or make a new blank row.

```
  \LT@make@row
```

A few more internal commands for `longtable`.

```
  \let\@endpbox\LT@endpbox
  \m@th\let\par\@empty
  \everycr{}\lineskip\z@\baselineskip\z@
```

Start the first chunk.

```
  \LT@bchunk}
```

`\LT@start` This macro starts the process of putting the table on the current page. It is not called until either a `\\` or `\endlongtable` command ends a chunk, as we do not know until that point which of the four possible head or foot sections have been specified.

It begins by redefining itself, so that the table is only started once! Until V3.04, was redefined to `\relax`, now use `\endgraf` to force the page-breaker to wake up.

```
\def\LT@start{%
  \let\LT@start\endgraf
  \endgraf\penalty\z@\vskip\LTpre
```

Start a new page if there is not enough room for the table head, foot, and one extra line.

```
  \dimen@\pagetotal
  \advance\dimen@ \ht\ifvoid\LT@firsthead\LT@head\else\LT@firsthead\fi
  \advance\dimen@ \dp\ifvoid\LT@firsthead\LT@head\else\LT@firsthead\fi
  \advance\dimen@ \ht\LT@foot
  \advance\dimen@ \ht\@arstrutbox
  \advance\dimen@ \dp\@arstrutbox
  \advance\dimen@ -\pagegoal
  \ifdim \dimen@>\z@\vfil\break\fi
```

Store height of page minus table foot in `\@colroom`.

```
      \global\@colroom\@colht
```

If the foot is non empty, reduce the `\vsize` and `\@colroom` accordingly.

```
  \ifvoid\LT@foot\else
    \advance\vsize-\ht\LT@foot
    \global\advance\@colroom-\ht\LT@foot
    \dimen@\pagegoal\advance\dimen@-\ht\LT@foot\pagegoal\dimen@
```

```
    \maxdepth=\z@
  \fi
```

Put the table head on the page, and then switch to the new output routine.

```
  \ifvoid\LT@firsthead\copy\LT@head\else\box\LT@firsthead\fi
  \output{\LT@output}}
```

\endlongtable Called by \end{longtable}.

```
\def\endlongtable{%
```

Essentially add a final \\.

```
  \LT@echunk
  \LT@start
  \unvbox\z@
  \LT@get@widths
```

Write the dummy row to the .aux file. Since V3.06, use .lta for letter.sty.

```
  \if@filesw\immediate\write\@auxout
```

Since Version 3.02, longtable has used the internal counter \c@LT@tables rather than the LaTeX counter table.

```
    {\gdef\string\LT@\romannumeral\c@LT@tables{\LT@save@row}}\fi
```

Issue a warning if a \multicolumn has been set in draft mode.

```
  \LT@mcw@rn
```

If the last chunk has different widths than the first, warn the user.

```
  \ifx\LT@save@row\LT@@save@row\else
    \LT@warn{Column widths have changed\MessageBreak
             in table \thetable}\fi
```

Force one more go with the longtable ouput routine.

```
  \endgraf\penalty -\LT@end@pen
```

Now close the group to return to the standard routine.

```
  \endgroup
```

Reset \@mparbottom to allow marginpars close to the end of the table.[5]

```
  \global\@mparbottom\z@
  \pagegoal\vsize
  \endgraf\penalty\z@\addvspace\LTpost
```

Footnotes. As done in multicol.sty.

```
  \ifvoid\footins\else\insert\footins{}\fi}
```

9.6 Counting Columns

Columns are counted by examining \@preamble, rather than simply getting \@mkpream to increment the counter as it builds the preamble so that this package works with many of the packages which add extra column specifiers to LaTeX's standard ones.

Version 1 counted \@sharp's to calculate the number of columns, this was changed for Version 2 as it does not work with the NFSS. Now count &'s. (lfonts.new defines \@tabclassz so that \@sharp is inside a group.)

[5] This can not be the correct. However if it is omitted, there is a problem with marginpars, for example on page 3 of this document. Any Output Routine Gurus out there?

\LT@nofcols Find the next &, then look ahead to see what is next.

```
\def\LT@nofcols#1&{\futurelet\LT@next\LT@n@fcols}
```

\LT@n@fcols Add one, then stop at an \LT@nofcols or look for the next &. The \expandafter trick was added in Version 3, also the name changed from \@LT@nofcols to preserve the \LT@ naming convention.

```
\def\LT@n@fcols{\advance\LT@cols\@ne
  \ifx\LT@next\LT@nofcols\expandafter\@gobble
  \else\expandafter\LT@nofcols\fi}
```

9.7 The \\ and \kill Commands

\LT@tabularcr The internal definition of \\.

```
\def\LT@tabularcr{%
```

Increment the counter, and do tabular's \\ or finish the chunk.
The \expandafter trick was added in Version 3.

```
  \global\advance\LT@rows\@ne \ifnum\LT@rows=\LTchunksize
  \expandafter\LT@t@bularcr\else\expandafter\LT@LL@FM@cr\fi}
```

\LT@t@bularcr This definition also accepts *, which acts in the same way as \\. tabular does this, but longtable probably ought to make * prevent page breaking. {\ifnum0=`}\fi added in version 3.01, required if the first entry is empty. The above in fact is not good enough, as with array package it can introduce a {} group in math mode, which changes the spacing. So use the following variant. Added in v3.14.

```
\def\LT@t@bularcr{\relax\iffalse{\fi\ifnum0=`}\fi
  \@ifstar{\LT@xtabularcr}{\LT@xtabularcr}}
```

\LT@xtabularcr This justs looks for an optional argument.

```
\def\LT@xtabularcr{\@ifnextchar[{\LT@argtabularcr}{\LT@ntabularcr}}
```

\LT@ntabularcr The version with no optional argument. \ifnum0=`{\fi} added in version 3.01. Changed in 3.14.

```
\def\LT@ntabularcr{%
  \ifnum0=`{}\fi
  \LT@echunk
  \LT@start
  \unvbox\z@
  \LT@get@widths
  \LT@bchunk}
```

\LT@argtabularcr The version with an optional argument. \ifnum0=`{\fi} added in version 3.01. Changed in 3.14.

```
\def\LT@argtabularcr[#1]{%
  \ifnum0=`{}\fi
  \ifdim #1>\z@
  \unskip\@xargarraycr{#1}\else \@yargarraycr{#1}\fi
```

Add the dummy row, and finish the \halign.

```
  \LT@echunk
  \LT@start
  \unvbox\z@
```

```
  \LT@get@widths
  \LT@bchunk}
```

`\LT@echunk` This ends the current chunk, and removes the dummy row.

```
\def\LT@echunk{%
  \crcr\LT@save@row\cr\egroup
  \global\setbox\@ne\lastbox
  \egroup}
```

`\LT@kill` This ends the current chunk as above, but strips off two rows, the 'dummy row' and the 'killed row' before starting the next chunk. Since V3.04, the old chunk is reboxed at the start of the box containing the next chunk. This allows `\kill` to be used in headers, which must be processed in a single box.

```
\def\LT@kill{%
  \LT@echunk
  \LT@get@widths
  \expandafter\LT@rebox\LT@bchunk}
```

`\LT@rebox` Drop the old chunk (box0) back at the top of the new chunk, removing the killed row. This macro added at V3.04.

```
\def\LT@rebox#1\bgroup{#1\bgroup\unvbox\z@\unskip\setbox\z@=\lastbox}%

```

9.8 The Dummy Row

`\LT@blank@row` Create a blank row if we are not using the info in the .aux file. Also use the draft form of `\multicolumn`.

```
\def\LT@blank@row{%
  \let\multicolumn\LTmulticolumn
    {\xdef\LT@save@row{\omit}%
    \loop
    \advance\LT@cols\m@ne
    \ifnum\LT@cols>\z@
    \xdef\LT@save@row{\LT@save@row&\omit}%
    \repeat}}
```

`\LT@make@row` By default do not use information in the `.aux` file.

```
\let\LT@make@row\LT@blank@row
```

`\setlongtables` Redefine `\LT@make@row` to use information in the `.aux` file, if there is a saved row for this table.

Since Version 3.02, `longtable` has used the internal counter `\c@LT@tables` rather than the LaTeX counter `table`. The warning message was added at V3.04, as was the `\global`, to stop save-stack overflow.

```
\def\setlongtables{%
  \LT@warn{%
    \noexpand\setlongtables is in force.%
    \MessageBreak
    If you have edited the file since the last%
    \MessageBreak
    LaTeX run, incorrect column widths, or even%
    \MessageBreak
```

```
    errors, may be produced.%
    \@gobbletwo}%
  \def\LT@make@row{%
    \global\expandafter\let\expandafter\LT@save@row
      \csname LT@\romannumeral\c@LT@tables\endcsname
    \ifx\LT@save@row\relax\LT@blank@row\fi}}
```

`\LT@get@widths` This is the heart of `longtable`. If it were not for the table head and foot, this macro together with the modified `\\` command would form the basis of quite a simple little package file for long tables. It is closely modelled on the `\endvrulealign` macro of appendix D of the TEXBook.

```
\def\LT@get@widths{%
```

`\global` added at V3.04, to stop save-stack overflow.

```
  \global\let\LT@save@row\end
```

Loop through the last row, discarding glue, and saving box widths. At V3.04 changed the scratch box to 2, as the new `\kill` requires that `\box0` be preserved.

```
  \setbox\tw@=\hbox{\unhbox\@ne
    \loop
    \unskip
    \setbox\tw@=\lastbox
    \ifhbox\tw@
      \LT@def@row
    \repeat}%
```

Remember the widths if we are in the first chunk.

```
  \ifx\LT@@save@row\LT@undef\let\LT@@save@row\LT@save@row\fi}
```

`\LT@def@row` Add a column to the dummy row. Name changed from `\defLT@save@row` in Version 3, to preserve the `\LT@` naming convention.

```
\def\LT@def@row{\ifx\LT@save@row\end
  \xdef\LT@save@row{\omit\kern\the\wd\tw@}\else
  \xdef\LT@save@row{\omit\kern\the\wd\tw@&\LT@save@row}\fi}
```

9.9 The `\hline` Command

`\LT@hline` `\hline` and `\hline\hline` both produce *two* lines. The only difference being the glue and penalties between them. This is so that a page break at a `\hline` produces a line on both pages.[6] Also this `\hline` is more like a `\cline[1-\LT@cols]`. `tabular`'s `\hline` would draw lines the full width of the page.

```
\def\LT@hline{\noalign{\ifnum0=`}\fi
   \penalty\@M
   \futurelet\LT@next\LT@@hline}
```

`\LT@@hline` This code is based on `\cline`. Two copies of the line are produced, as described above.

```
\def\LT@@hline{%
  \ifx\LT@next\hline
    \global\let\LT@next\@gobble
    \gdef\LT@sep{\penalty-\@medpenalty\vskip\doublerulesep}%
```

[6] `longtable` has always done this, but perhaps it would be better if hlines were *omitted* at a page break, as the head and foot usually put a hline here anyway.

```
  \else
    \global\let\LT@next\empty
    \gdef\LT@sep{\penalty-\@lowpenalty\vskip-\arrayrulewidth}%
  \fi\ifnum0=`{\fi}%
  \multispan\LT@cols
     \unskip\leaders\hrule\@height\arrayrulewidth\hfill\cr
  \noalign{\LT@sep}%
  \multispan\LT@cols
     \unskip\leaders\hrule\@height\arrayrulewidth\hfill\cr
  \noalign{\penalty\@M}%
  \LT@next}
```

9.10 Captions

`\LT@caption` The caption is `\multicolumn{\LT@cols}{c}{`⟨*a parbox with the table's caption*⟩`}`

```
\def\LT@caption{\noalign\bgroup
  \@ifnextchar[{\egroup\LT@c@ption\@firstofone}\LT@capti@n}
```

`\LT@c@ption` Caption command (with [optional argument]). `\protect` added in Version 3. `\fnum@table` added at V3.05.

```
\def\LT@c@ption#1[#2]#3{%
  \LT@makecaption#1\fnum@table{#3}%
  \def\@tempa{#2}\ifx\@tempa\@empty\else
     {\let\\\space
     \addcontentsline{lot}{table}{\protect\numberline{\thetable}{#2}}}%
  \fi}
```

`\LT@capti@n` Caption command (no [optional argument])

```
\def\LT@capti@n{%
  \@ifstar
    {\egroup\LT@c@ption\@gobble[]}%
    {\egroup\@xdblarg{\LT@c@ption\@firstofone}}}
```

`\LT@makecaption` Put the caption in a box of width 0pt, so that it never affects the column widths. Inside that is a `\parbox` of width `\LTcapwidth`.

```
\def\LT@makecaption#1#2#3{%
  \LT@mcol\LT@cols c{\hbox to\z@{\hss\parbox[t]\LTcapwidth{%
```

Based on article class `\@makecaption`, `#1` is `\@gobble` in star form, and `\@firstofone` otherwise.

```
    \sbox\@tempboxa{#1{#2: }#3}%
    \ifdim\wd\@tempboxa>\hsize
      #1{#2: }#3%
    \else
      \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
    \fi
    \endgraf\vskip\baselineskip}%
  \hss}}}
```

9.11 The Output Routine

The method used here for interfacing a special purpose output routine to the standard LaTeX routine is lifted straight out of F. Mittelbach's `multicol.sty`.

`\LT@output` Actually this is not so bad, with FM leading the way.

```
\def\LT@output{%
  \ifnum\outputpenalty <-\@Mi
    \ifnum\outputpenalty > -\LT@end@pen
```

If this was a float or a marginpar we complain.

```
      \LT@err{floats and marginpars not allowed in a longtable}\@ehc
    \else
```

We have reached the end of the table, on the scroll at least,

```
      \setbox\z@\vbox{\unvbox\@cclv}%
      \ifdim \ht\LT@lastfoot>\ht\LT@foot
```

The last foot might not fit, so:[7]

```
        \dimen@=\pagegoal
        \advance\dimen@-\ht\LT@lastfoot
        \ifdim\dimen@<\ht\z@
          \setbox\@cclv=\vbox{\unvbox\z@\copy\LT@foot\vss}%
          \@makecol
          \@outputpage
          \setbox\z@\vbox{\box\LT@head}%
```

End of `\ifdim\dimen@<\ht\@cclc`.

```
        \fi
```

End of `\ifdim \ht\LT@lastfoot > \ht\LT@foot`.

```
      \fi
```

Reset `\@colroom`.

```
      \global\@colroom\@colht
      \global\vsize\@colht
```

Put the last page of the table on to the main vertical list.

```
      \vbox
        {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
```

End of `\ifnum\outputpenalty > -\LT@end@pen`.

```
    \fi
```

Else `\outputpenalty > -\@Mi`.

```
  \else
```

If we have not reached the end of the table,

```
    \setbox\@cclv=\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
    \@makecol
    \@outputpage
```

Reset `\vsize`.

```
      \global\vsize\@colroom
```

Put the head at the top of the next page.

```
    \copy\LT@head
```

End of `\ifnum\outputpenalty <-\@Mi`.

```
  \fi}
```

[7] An alternative would be to vsplit off a bit of the last chunk, so that the last page did not just have head and foot sections, but it is hard to do this in a consistent manner.

9.12 Commands for the the table head and foot

\LT@end@hd@ft

The core of `\endhead` and friends. Store the current chunk in the box specified by `#1`. Issue an error if the table has already started. Then start a new chunk.

```
\def\LT@end@hd@ft#1{%
  \LT@echunk
```

Changed from `\relax` to `\endgraf` at V3.04, see `\LT@start`.

```
  \ifx\LT@start\endgraf\LT@err
   {Longtable head or foot not at start of table}%
   {Increase LTchunksize}\fi
  \setbox#1\box\z@
  \LT@get@widths
  \LT@bchunk}
```

\endfirsthead
\endhead
\endfoot
\endlastfoot

Call `\LT@end@hd@ft` with the appropriate box.

```
\def\endfirsthead{\LT@end@hd@ft\LT@firsthead}
\def\endhead{\LT@end@hd@ft\LT@head}
\def\endfoot{\LT@end@hd@ft\LT@foot}
\def\endlastfoot{\LT@end@hd@ft\LT@lastfoot}
```

9.13 The `\multicolumn` command

\LTmulticolumn

A temporary definition of `\multicolumn` to use if we are ignoring the saved row in the `.aux` file. Hide the width of the entry with `\hidewidth\null`, and warn the user at the end that this 'draft' version of `\multicolumn` has been used.

```
\def\LTmulticolumn#1#2#3{%
  \LT@mcol{#1}{#2}{#3}\hidewidth\null
  \global\let\LT@mcw@rn\LT@mcwarn}
```

\LT@mcwarn

Typeout a warning if `\LTmulticolumn` has been used.

```
\def\LT@mcwarn{\LT@warn{%
  \string\multicolumn\space set in draft mode\MessageBreak
   in table \thetable}}
```

9.14 Footnotes

The standard `\footnote` command works in a `c` column, but we need to modify the definition in a `p` column to overcome the extra level of boxing. These macros are based on `array.sty`, but should be OK for the standard `tabular`.

\LT@startpbox

Add extra code to switch the definition of `\@footnotetext`.

```
\def\LT@startpbox#1{\bgroup
  \let\@footnotetext\LT@p@ftntext\hsize #1 \@arrayparboxrestore
  \vrule \@height \ht\@arstrutbox \@width \z@}
```

\LT@endpbox

After the parbox is closed, expand `\LT@p@ftn` which will execute a series of
`\footnotetext[`⟨*num*⟩`]{`⟨*note*⟩`}`
commands. After being lifted out of the parbox, they can migrate on their own from here.

```
\def\LT@endpbox{\@finalstrut\@arstrutbox\egroup
      \the\LT@p@ftn\global\LT@p@ftn{}}
```

\LT@p@ftntext Inside the ‘p’ column, just save up the footnote text in a token register.

```
\def\LT@p@ftntext#1{%
  \edef\@tempa{\the\LT@p@ftn\noexpand\footnotetext[\the\c@footnote]}%
  \global\LT@p@ftn\expandafter{\@tempa{#1}}}%
```

⟨/package⟩