The longtable package*

David Carlisle†

2004/02/01

Abstract

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

List of Tables

1 An optional table caption (used in the list of tables) ............ 2
2 A floating table .............................................. 4
3 A difficult \multicolumn combination: pass 1 .................. 6
4 A difficult \multicolumn combination: pass 2 .................. 6
5 A difficult \multicolumn combination: pass 3 .................. 6
6 A difficult \multicolumn combination: pass 4 .................. 6
7 A summary of longtable commands .............................. 9

1 Introduction

The longtable package 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 until this document has been run through \LaTeX several times. This is a characteristic feature of this package, as described below.

---

*This file has version number v4.11, last revised 2004/02/01.
†The new algorithm for aligning ‘chunks’ of a table used in version 4 of this package was devised coded and documented by David Kastrup, dak@neuroinformatik.ruhr-uni-bochum.de.
Table 1: A long table

<table>
<thead>
<tr>
<th>FIRST</th>
<th>SECOND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longtable columns are specified in the same way as in the <code>\tabular</code> environment.</td>
<td></td>
</tr>
<tr>
<td>Each row ends with a <code>\\</code> command.</td>
<td></td>
</tr>
<tr>
<td>The <code>\\</code> command has an optional argument, just as in the <code>\tabular</code> environment.</td>
<td></td>
</tr>
<tr>
<td>See the effect of <code>\[10pt]</code></td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines as in <code>\tabular</code>.</td>
<td></td>
</tr>
<tr>
<td>That was a <code>\hline</code>.</td>
<td></td>
</tr>
<tr>
<td>This is a `\multicolumn{2}{</td>
<td></td>
</tr>
<tr>
<td>If a page break occurs at a <code>\hline</code> then a line is drawn at the bottom of one page and at the top of the next.</td>
<td></td>
</tr>
<tr>
<td>The <code>[t] [b] [c]</code> argument of <code>\tabular</code> can not be used.</td>
<td></td>
</tr>
<tr>
<td>The optional argument may be one of <code>[l] [r] [c]</code> adjusted or centrally.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>Lots of lines like this.</td>
<td></td>
</tr>
<tr>
<td>This goes at the bottom.</td>
<td></td>
</tr>
</tbody>
</table>
Table 1: (continued)

<table>
<thead>
<tr>
<th>*</th>
<th>This part appears at the top of every other page</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td><strong>First</strong></td>
</tr>
</tbody>
</table>
| * | *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 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.  
| * | These lines will appear in place of the usual foot of the table.
| * | at the end

2 Chunk Size

\texttt{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., `\setcounter{LTchunksize}{10}`. These chunks do not affect page breaking, thus if you are using a \TeX with a lot of memory, you can set \texttt{LTchunksize} to be several pages of the table. \TeX will run faster with a large \texttt{LTchunksize}.

\footnote{This is a footnote.}
\footnote{longtable takes special precautions, so that footnotes may also be used in ‘p’ columns.}
\footnote{You can also use the plain \TeX syntax `\LTchunksize=10`.}
Table 2: A floating table

However, if necessary, longtable can work with \LTchunksize set to 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 \setcounter{LTchunksize}{10}. If you look at the previous table, after the first run of \LaTeX you will see that various parts of the table do not line up. \LaTeX will also 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. Prior to version 4 of this package, this information was not used unless a \setlongtables command was issued, however, now the information is always used, using a new algorithm\footnote{Due to David Kastrup.} and so \setlongtables is no longer needed. It is defined (but does nothing) for the benefit of old documents that use it.

## 3 Captions and Headings

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.

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}{2in} 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 table caption (used in 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 \texttt{\caption*} command.

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

A more convenient method of customising captions is given by the \texttt{caption(2)} package, which provides commands for customising captions, and arranges that the captions in standard environments, and many environments provided by packages (including \texttt{longtable}) are modified in a compatible manner.

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

4 Multicolumn entries

The \texttt{\multicolumn} command may be used in \texttt{longtable} in exactly the same way as for \texttt{tabular}. So you may want to skip this section, which is rather technical, however coping with \texttt{\multicolumn} is one of the main problems for an environment such as \texttt{longtable}. The main effect that a user will see is that certain combinations of \texttt{\multicolumn} entries will result in a document needing more runs of \LaTeX{} before the various ‘chunks’ of a table align.

The examples in this section are set with \texttt{\LTchunksize} set to the minimum value of one, to demonstrate the effects when \texttt{\multicolumn} entries occur in different chunks.

Consider Table 3. In the second chunk, \texttt{longtable} sees the wide multicolumn entry. At this point it thinks that the first two columns are very narrow. All the width of the multicolumn entry is assumed to be in the third column. (This is a ‘feature’ of \TeX{}’s primitive \texttt{\halign} command.) \texttt{longtable} then passes the information that there is a wide third column to the later chunks, with the result that the first pass over the table is too wide.

If the ‘saved row’ from this first pass was re-inserted into the table on the next pass, the table would line up in two passes, but would be much too wide.

\begin{verbatim}
\kill
\end{verbatim}

The solution to this problem used in Versions 1 and 2, was to use a \texttt{\kill} line. If a line is \texttt{\kill}ed, by using \texttt{\kill} rather than \texttt{\} at the end of the line, it is used in calculating column widths, but removed from the final table. Thus entering \texttt{\kill}ed copies of the last two rows before the wide multicolumn entry would mean that \texttt{\halign} ‘saw’ the wide entries in the first two columns, and so would not widen the third column by so much to make room for the multicolumn entry.

In Version 3, a new solution was introduced. If the saved row in the \texttt{.aux} file was not being used, \texttt{longtable} used a special ‘draft’ form of \texttt{\multicolumn}, this modified the definition, so the spanning entry was never considered to be wider than the columns it spanned. So after the first pass, the \texttt{.aux} file stored the widest normal entry for each column, no column was widened due to \texttt{spanned} columns. By default \texttt{longtable} ignored the \texttt{.aux} file, and so each run of \LaTeX{} was considered a first pass. Once the \texttt{\setlongtables} declaration was given, the saved row in the \texttt{.aux} file, and the proper definition of \texttt{\multicolumn} were...
Table 3: A difficult \texttt{\textbackslash multicolumn} combination: pass 1

\begin{tabular}{|c|c|c|}
\hline
1 & 2 & 3 \\
wide multicolumn spanning 1–3 & & \\
multicolumn 1–2 & 3 & \\
wide 1 & 2 & 3 \\
\hline
\end{tabular}

Table 4: A difficult \texttt{\textbackslash multicolumn} combination: pass 2

\begin{tabular}{|c|c|c|}
\hline
1 & 2 & 3 \\
wide multicolumn spanning 1–3 & & \\
multicolumn 1–2 & 3 & \\
wide 1 & 2 & 3 \\
\hline
\end{tabular}

Table 5: A difficult \texttt{\textbackslash multicolumn} combination: pass 3

\begin{tabular}{|c|c|c|}
\hline
1 & 2 & 3 \\
wide multicolumn spanning 1–3 & & \\
multicolumn 1–2 & 3 & \\
wide 1 & 2 & 3 \\
\hline
\end{tabular}

Table 6: A difficult \texttt{\textbackslash multicolumn} combination: pass 4

\begin{tabular}{|c|c|c|}
\hline
1 & 2 & 3 \\
wide multicolumn spanning 1–3 & & \\
multicolumn 1–2 & 3 & \\
wide 1 & 2 & 3 \\
\hline
\end{tabular}
used. If any \multicolumn entry caused one of the columns to be widened, this 
information could not be passed back to earlier chunks, and so the table would not 
correctly line up until the third pass. This algorithm always converged in three 
passes as described above, but in examples such as the ones in Tables 3–6, the 
final widths were not optimal as the width of column 2, which is determined by a 
\multicolumn entry was not known when the final width for column 3 was fixed, 
due to the fact that both \multicolumn commands were switched from ‘draft’ 
mode to ‘normal’ mode at the same time.

Version 4 alleviates the problem considerably. The first pass of the table will 
indeed have the third column much too wide. However, on the next pass \longtable 
will notice the error and reduce the column width accordingly. If this has to 
propagate to chunks before the \multicolumn one, an additional pass will, of 
course, be needed. It is possible to construct tables where this rippling up of 
the correct widths takes several passes to ‘converge’ and produce a table with all 
chunks aligned. However in order to need many passes one needs to construct a table with 
many overlapping \multicolumn entries, all being wider than the natural widths 
of the columns they span, and all occuring in different chunks. In the typical 
case the algorithm will converge after three or four passes, and, the benefits of 
ot needing to edit the document before the final run to add \setlongtables, 
and the better choice of final column widths in the case of multiple \multicolumn 
entries will hopefully more than pay for the extra passes that may possibly be 
needed.

So Table 3 converges after 4 passes, as seen in Table 6.

You can still speed the convergence by introducing judicious \kill lines, if you 
happen to have constellations like the above.

If you object even to \LaTeX-ing a file twice, you should 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.

5 Adjustment

The optional argument of \longtable controls the horizontal alignment of the table. 
The possible options are \[\{c\}, \[r\} and \[l\], for centring, right and left adjustment, 
respectively. Normally centring is the default, but this document specifies

\[\begin{longtable}{@{\extracolsep{...}}...} \end{longtable}\]

in the preamble, 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. For instance

\[\begin{tabular*}{\textwidth}{\@{\extracolsep{...}}}...\end{tabular*}\]

produces a full width table, to get a similar effect with \longtable specify

\[\begin{longtable}{@{\extracolsep{...}}}...\end{longtable}\]
6 Changes

This section highlights the major changes since version 2. A more detailed change log may be produced at the end of the code listing if the \texttt{ltxdoc.cfg} file specifies

\begin{verbatim}
\AtBeginDocument{\RecordChanges}
\AtEndDocument{\PrintChanges}
\end{verbatim}

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, \texttt{longtable} does not need to issue a \texttt{\clearpage} at the start of the table, and so the table may start half way down a page. Also the \texttt{\endlastfoot} command which could not safely be implemented under the old scheme, has been added.

- \texttt{longtable} now issues an error if started in the scope of \texttt{\twocolumn}, or the \texttt{multicol} environment.

- The separate documentation file \texttt{longtable.tex} has been merged with the package file, \texttt{longtable.dtx} using Mittelbach’s \texttt{doc} package.

- Support for footnotes has been added. Note however that \texttt{\footnote} will not work in the ‘head’ or ‘foot’ sections of the table. In order to put a footnote in those sections (e.g., inside a caption), use \texttt{\footnotemark} at that point, and \texttt{\footnotetext} anywhere in the table body that will fall on the same page.

- The treatment of \texttt{\multicolumn} has changed, making \texttt{\kill} lines unnecessary, at the price of sometimes requiring a third pass through \texttt{\LaTeX}.

- The \texttt{\newpage} command now works inside a \texttt{longtable}.

Changes made between versions 3 and 4.

- A new algorithm is used for aligning chunks. As well as the widest width in each column, \texttt{longtable} remembers which chunk produced this maximum. This allows it to check that the maximum is still achieved in later runs. As \texttt{longtable} can now deal with columns shrinking as the file is edited, the \texttt{\setlongtables} system is no longer needed and is disabled.

- An extra benefit of the new algorithm’s ability to deal with ‘shrinking’ columns is that it can give better (narrower) column widths in the case of overlapping \texttt{\multicolumn} entries in different chunks than the previous algorithm produced.

- The ‘draft’ \texttt{\multicolumn} system has been removed, along with related commands such as \texttt{\LTmulticolumn}.

- The disadvantage of the new algorithm is that it can take more passes. The theoretical maximum is approximately twice the length of a ‘chain’ of columns with overlapping \texttt{\multicolumn} entries, although in practice it usually converges as fast as the old version. (Which always converged in three passes once \texttt{\setlongtables} was activated.)

- \texttt{\\*} and \texttt{\nopagebreak} commands may be used to control page breaking.
7 Summary

Table 7: A summary of \texttt{longtable} commands

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\LTleft</td>
<td>Glue to the left of the table. \texttt{(\fill)}</td>
</tr>
<tr>
<td>\LTright</td>
<td>Glue to the right of the table. \texttt{(\fill)}</td>
</tr>
<tr>
<td>\LTpre</td>
<td>Glue before the table. \texttt{(~\texttt{\bigskipamount})}</td>
</tr>
<tr>
<td>\LTpost</td>
<td>Glue after the table. \texttt{(~\texttt{\bigskipamount})}</td>
</tr>
<tr>
<td>\LTcapwidth</td>
<td>The width of a parbox containing the caption. \texttt{(~4in)}</td>
</tr>
<tr>
<td>\LTchunksize</td>
<td>The number of rows per chunk. \texttt{(~20)}</td>
</tr>
</tbody>
</table>

Optional arguments to \texttt{\begin{longtable}}


<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Position as specified by \LTleft and \LTright.</td>
</tr>
<tr>
<td>[c]</td>
<td>Centre the table.</td>
</tr>
<tr>
<td>[l]</td>
<td>Place the table flush left.</td>
</tr>
<tr>
<td>[r]</td>
<td>Place the table flush right.</td>
</tr>
</tbody>
</table>

Commands to end table rows

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\</td>
<td>Specifies the end of a row</td>
</tr>
<tr>
<td>\[\texttt{\langle dim\rangle}]</td>
<td>Ends row, then adds vertical space (as in the \texttt{tabular} environment).</td>
</tr>
<tr>
<td>\*</td>
<td>The same as \ but disallows a page break after the row.</td>
</tr>
<tr>
<td>\tabularnewline</td>
<td>Alternative to \ for use in the scope of \texttt{\raggedright} and similar commands that redefine \.</td>
</tr>
<tr>
<td>\kill</td>
<td>Row is ‘killed’, but is used in calculating widths.</td>
</tr>
<tr>
<td>\endhead</td>
<td>Specifies rows to appear at the top of every page.</td>
</tr>
<tr>
<td>\endfirsthead</td>
<td>Specifies rows to appear at the top the first page.</td>
</tr>
<tr>
<td>\endfoot</td>
<td>Specifies rows to appear at the bottom of every page.</td>
</tr>
<tr>
<td>\endlastfoot</td>
<td>Specifies rows to appear at the bottom of the last page.</td>
</tr>
</tbody>
</table>

\texttt{longtable} caption commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\caption{\langle caption\rangle}</td>
<td>Caption ‘Table ?: \langle caption\rangle’, and a ‘\langle caption\rangle’ entry in the list of tables.</td>
</tr>
<tr>
<td>\caption{\langle lot\rangle}{\langle caption\rangle}</td>
<td>Caption ‘Table ?: \langle caption\rangle’, and a ‘\langle lot\rangle’ entry in the list of tables.</td>
</tr>
<tr>
<td>\caption{}{\langle caption\rangle}</td>
<td>Caption ‘Table ?: \langle caption\rangle’, but no entry in the list of tables.</td>
</tr>
<tr>
<td>\caption*{\langle caption\rangle}</td>
<td>Caption ‘\langle caption\rangle’, but no entry in the list of tables.</td>
</tr>
</tbody>
</table>

Commands available at the start of a row

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\pagebreak</td>
<td>Force a page break.</td>
</tr>
<tr>
<td>\pagebreak[\langle val\rangle]</td>
<td>A ‘hint’ between 0 and 4 of the desirability of a break.</td>
</tr>
<tr>
<td>\nopagebreak</td>
<td>Prohibit a page break.</td>
</tr>
<tr>
<td>\nopagebreak[\langle val\rangle]</td>
<td>A ‘hint’ between 0 and 4 of the undesirability of a break.</td>
</tr>
<tr>
<td>\newpage</td>
<td>Force a page break.</td>
</tr>
</tbody>
</table>

Footnote commands available inside \texttt{longtable}

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\footnote</td>
<td>Footnotes, but may not be used in the table head &amp; foot.</td>
</tr>
<tr>
<td>\footnotemark</td>
<td>Footnotemark, may be used in the table head &amp; foot.</td>
</tr>
<tr>
<td>\footnotetext</td>
<td>Footnote text, use in the table body.</td>
</tr>
</tbody>
</table>

\texttt{Setlongtables}

\setlongtables | Obsolete command. Does nothing now. |
\begin{longtable}{@{}r||p{1in}@{}}
\caption{An optional table caption ...}
\hline
\multicolumn{2}{@{}c@{}}{This part appears at the top of the table}\
\textsc{First}&\textsc{Second}\
\hline\hline
\caption{Continued}\
\multicolumn{2}{@{}c@{}}{This part appears at the top of every other page}\
\textbf{First}&\textbf{Second}\
\hline
\endhead
\hline
This goes at the bottom.\
\hline
\hline
These lines will appear in place of the & usual foot\
\hline
\endfoot
\hline
\hline
\env{longtable} columns are specified in the \env{tabular} & environment.\ & \...
\multicolumn{2}{r}{This is a ...}\ & \...
\multicolumn{2}{c}{This last column is a 'p' column...}
\tabularnewline
\hline
Lots of lines \footnote{...} of lines like this.\
\footnote{...} of lines like this\
\hline
Lots of lines like this.\
\hline
Lots of lines like this.\
\hline
\hline
\end{longtable}
9 The Macros

9.1 Initial code

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

\LT@err The error generating command
\LT@err\PackageError{longtable}

\LT@warn The warning generating command
\LT@warn\PackageWarning{longtable}

\LT@final@warn If any longtables have not aligned, generate a warning at the end of the run at \AtEndDocument.
\LT@final@warn{Table \width s have changed. Rerun \LaTeX. \gobbletwo}

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}{\LT@warn\PackageInfo{longtable}}

pausing Make every warning message into an error so \TeX stops. May be useful for debugging.
\DeclareOption{pausing}{\LT@err#1{This is not really an error}}

set The next options are just alternative syntax for the \setlongtables declaration.
\DeclareOption{set}{\LTleft=\LTfill\LTpost=\bigskipamount}

final

9.3 User Setable Parameters

\LTleft Glue to the left and right of the table, default \fill (ie centred).
\LTleft=\fill
\LTleft=\fill
\LTright Glue before and after the longtable. \bigskip by default.
\LTpre=\bigskipamount
\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 \TeX\ 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 Boxes for the table head and foot.
\LT@firsthead
\LT@foot
\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 \TeX\ 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. \TeX\ 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.
\newcounter{LT@tables}
\c@LT@chunks We need to count through the chunks of our tables from Version 4 on.
\newcounter{LT@chunks}[LT@tables]
\c@table 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}\%
\longtable
\begin{longtable}

\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>1
\@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 the \LaTeX kernel and the array package.

Since Version 3.02, \longtable has used the internal counter \c@LT@tables. The \LaTeX counter \texttt{table} is still incremented so that \texttt{caption} works correctly.
\def\LT@array[#1]{%
\refstepcounter{table}\stepcounter{LT@tables}%
\set up the glue around the table if an optional argument given.
\if #1%
\LTleft\z@ \LTright\fill
\else\if r#1%
\LTleft\fill \LTright\z@%
\else\if c#1%
\LTleft\fill \LTright\fill
\fi\fi\fi
\fi\fi\fi

Set up these internal commands for \longtable.
\global\let\LT@mcol\multicolumn

\}
Now redefine \tabarray to restore \hline and \multicolumn so that arrays and tabulars nested in longtable (or in page headings on longtable pages) work out OK. Saving the original definitions done here so that you can load the array package before or after longtable.

\let\LT@tabarray\@tabarray
\let\LT@hl\hline
\def\@tabarray{%
\let\hline\LT@hl
\let\multicolumn\LT@mcol%
\let\LT@@tabarray}
\let\LT@no@pgbk-\LT@no@pgbk
\def\newpage{%
\noalign{\break}%
More or less standard definitions, but first start a \noalign. 
\def\pagebreak{%
\noalign{\ifnum'}=0\fi\@testopt{\LT@no@pgbk-}4}%
\def\nopagebreak{%
\noalign{\ifnum'}=0\fi\@testopt\LT@no@pgbk4}%
\let\hline\LT@hline \let\kill\LT@kill\let\caption\LT@caption
\@tempdima\ht\strutbox
\let\@endpbox\LT@endpbox
Set up internal commands according to Lamport or Mittelbach.
\ifx\extrarowheight\@undefined
Initialise these commands as in tabular from the \LaTeX kernel.
\let\@acol\@tabacol
\let\@classz\@tabclassz \let\@classiv\@tabclassiv
\def\@startpbox{%
\vtop\LT@startpbox}%
\let\@@startpbox\@startpbox
\let\@@endpbox\@endpbox
\let\LT@LL@FM@cr\@tabularcr
\else
Initialise these commands as in array. \dollarsem replaced by \dollarsemb begin \dollarsemend in V3.03 to match array V2.0h. We do not need to set \dollarsembegin and \dollarsemend as the array package gives them the correct values at the top level.
\advance\@tempdima\extrarowheight
\col@sep\tabcolsep
\let\@startpbox\LT@startpbox\let\LT@LL@FM@cr\LT@arraycr
\fi
The rest of this macro is mainly based on array package, 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}%
\endgroup
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!
\def\LT@bchunk{
    \global\advance\c@LT@chunks\@one
    \global\LT@rows\z@\setbox\z@vbox{bgroup}
}

The following line was added in v4.05. In order to get the penalties to work at chunk boundaries Need to take more care about where and when \lineskip glue is added. The following does nothing at top of table, and in header chunks, but in normal body chunks it sets \prevdepth (to 0pt, but any value would do) so that \lineskip glue will be added. The important thing to note is that the glue will be added after any vertical material coming from \noalign.
\LT@setprevdepth
\tabskip\LTleft \noexpand\halign to\hsize\bgroup
% \tabskip\LTleft\halign to\hsize\bgroup
\LT@make@row
\m@th\let\par\@empty
\everycr{}\lineskip\z@\baselineskip\z@

Start the first chunk.
\LT@bchunk}
\LT@no@pgbk

Can simplify the standard \no@pgbk as this is vmode only but then need to close the \noalign.
\def\LT@no@pgbk#1[#2]{\penalty #1\@getpen{#2}\ifnum'{=0\fi}}

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.
\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.
At this point I used to add \ht\arstrutbox and \dp\arstrutbox as a measure of a row size. However this can fail spectacularly for p columns which might be much larger. Previous versions could end up with the table starting with a foot, then a page break then a head then a ‘first head’! So now measure the first line of the table accurately by \vsplitting it out of the first chunk.

\begin{longtable}
\end{longtable} Called by \end{longtable}.

\def\endlongtable{%
\end{longtable}

\endlongtable 

\end{longtable}

\end{longtable}
Since Version 3.02, longtable has used the internal counter \c@LT@tables rather than the \LaTeX counter table. This information looks entirely different from version 3 information. Still, we don’t need to rename the macro name because later code will consider the information to have no columns, and thus will throw the old data away.

\gdef\expandafter\noexpand
\csname LT@\romannumeral\c@LT@tables\endcsname
\{\LT@save@row\}\%
\fi

At this point used to 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. Also trigger a warning to rerun \LaTeX at the end of the document.

\ifx\LT@save@row\LT@@save@row
\else
\LT@warn{Column @width s have changed\MessageBreak
in table \thetable}\%
\LT@final@warn
\fi

Force one more go with the longtable output 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.\footnote{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?}

\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 (and now the Standard \LaTeX definition) defines \@tabclassz so that \@sharp is inside a group.)

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

\def\LT@nofcols\#1&{%
\LT@nofcols 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 pre-
serve the \LT@ naming convention.
\LT@tabularcr The internal definition of \\. In the * form, insert a \nobreak after the next \cr
(or \cr\cr).

This star form processing was finally added in v4.05. For the previous six or
seven years the comment at this point said
\LT@setprevdepth This will be redefined to set the \prevdepth at the start of a chunk.
\LT@t@bularcr \LT@setprevdepth\relax
\LT@t@bularcr Increment the counter, and do tabular's \ or finish the chunk.

The \expandafter trick was added in Version 3. Set the \prevdepth at the start
of a new chunk. (Done here so not set in header chunks).

9.7 The \ and \kill Commands
\LT@tabularcr The internal definition of \\.. In the * form, insert a \nobreak after the next \cr
(or \cr\cr).

This star form processing was finally added in v4.05. For the previous six or
seven years the comment at this point said
\LT@setprevdepth This will be redefined to set the \prevdepth at the start of a chunk.
\LT@t@bularcr \LT@setprevdepth\relax
\LT@t@bularcr Increment the counter, and do tabular's \ or finish the chunk.

The \expandafter trick was added in Version 3. Set the \prevdepth at the start
of a new chunk. (Done here so not set in header chunks).
\LT@xtabularcr This just 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
  \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
}\LT@echunk

\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
}\LT@echunk

The following line was added in v4.05. longtable relies on \lineskip glue (which is 0pt) to provide break points between each row so the table may be split into pages.

Previous releases left the \lineskip glue at the end of each chunk that had been added when the dummy row was added. There was no glue at the start of the next chunk as \TeX{} normally does not put \lineskip glue at the top of a box. This meant that normally the chunks fitted together perfectly, however \noalign material at a chunk boundary came before the first row of the next chunk but after the lineskip glue at the end of this chunk. This is the wrong place, e.g., it means even a \penalty10000 does not stop a break as the \lineskip glue in the previous item on the list provides a legal breakpoint. So now remove the \lineskip glue that was before the dummy row and introduce \LT@setprevdepth to set the \prevdepth at the start of the next chunk, to make sure \lineskip glue is added later.
\unskip
\egroup

\unskip
\egroup
\LT@entry We here give the ‘basic’ definition of \LT@entry, namely that used in alignment templates. It has a \kern only if the maximum is imposed from a different chunk. The \ifhmode test reveals the first entry, when we don’t want to add an &.
\LT@entry\@chop This definition for the argument of \LT@save@row is used to scrap all those maxima which could not be verified because they occur after the end of the table. This can happen only if a table has been shortened (or the sequencing got mixed up) since the previous run. Note that this is premature: the last chunk still is going to be set, and with the chopped limits.
\LT@entry\@write To write an entry for the aux file, we use a slightly surprising definition which has the sole purpose of avoiding overfull lines (which might break \TeX’s limits when reading the aux file, probably you’d need to have a few hundred columns before this happened but...).
\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.
\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.

9.8 The Dummy Row
The dummy row is kept inside of the macro \LT@save@row.
Create a blank row if we are not using the info in the .aux file.

Whoops! What's that supposed to be? A drop-in replacement for the first task of Appendix D in the \TeXbook. The \texttt{\romannumeral} produces \texttt{\LT@cols} instances of \texttt{m} followed by \texttt{i}. The below macro then replaces the \texttt{m}s by appropriate entries.

Prior to version 4, by default did not use information in the .aux file but now we can define \texttt{\LT@make@row} to use the .aux file, even on the ‘draft’ passes.

Now a slightly difficult part comes. Before we decide making the template from the .aux file info we check that the number of fields has remained the same. If it hasn’t, either the table format has changed, or we have the wrong table altogether. The expansion between \texttt{!}...! below will be empty if the number of \texttt{\LT@entry} macros including arguments in \texttt{\LT@save@row} is equal to \texttt{\LT@cols}. If it is not empty, we throw the row away and start from scratch.

Redefine \texttt{\LT@make@row} to use information in the .aux file, if there is a saved row for this table with the right number of columns.

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

Since Version 4.01 \texttt{\setlongtables} does nothing as it is not needed, but is defined as \texttt{\relax} for the benefit of old documents.
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 `\endrulalign` macro of appendix D of the TeXbook.

```
\def\LT@get@widths{%
  \global added at V3.04, to stop save-stack overflow.
  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
    \let\LT@old@row\LT@save@row
    \global\let\LT@save@row\@empty
    \count@\LT@cols
    \loop
      \unskip
      \setbox\tw@\lastbox
      \ifhbox\tw@\LT@def@row
      \advance\count@\m@ne
    \repeat}
  Remember the widths if we are in the first chunk.
  \ifx\LT@@save@row\@undefined
    \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{%
  We start by picking the respective entry from our old row. These redefinitions of `\LT@entry` are local to the group started in `\LT@get@widths`.
  \let\LT@entry\or
  \edef\@tempa{%
    \ifcase\expandafter\count@\LT@old@row
      \else
        \{1\}{0pt}%
    \fi}

  Now we tack the right combination in front of `\LT@save@row`:
  \let\LT@entry\relax
  \xdef\LT@save@row{%
    \LT@entry
    \expandafter\LT@max@sel\@tempa
    \LT@save@row}}
\LT@max@sel And this is how to select the right combination. Note that we take the old maximum information only if the size does not change in `either` direction. If the size has grown, we of course have a new maximum. If the size has shrunk, the old maximum (which was explicitly not enforced because of being in the current chunk) is invalid, and we start with this chunk as the new size. Note that even in the case of equality we must use the `\the\wd\tw@` construct instead of `#2` because `#2`
might be read in from the file, and so could have \catcode{11 versions of p and t in it which we want to be replaced by their ‘proper’ \catcode{12 versions.

\begin{verbatim}
def\LT@max@sel#1#2{\ifdim#2=\wd\tw@ \number\c@LT@chunks \else \number\c@LT@chunks \fi}{\the\wd\tw@}}
\end{verbatim}

9.9 The \texttt{\textbackslash hline} Command

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

\begin{verbatim}
def\LT@hline{% \noalign{\ifnum0='}{\fi} \penalty\@M \futurelet\@let@token\LT@@hline}
def\LT@@hline{\ifx\@let@token\hline \global\let\@gtempa\@gobble \gdef\LT@sep{\penalty-\@medpenalty\vskip\doublerulesep} \else \global\let\@gtempa\@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} \@gtempa}
\end{verbatim}

9.10 Captions

\texttt{\textbackslash LT@caption} The caption is \texttt{\multicolumn{\LT@cols}{c}{\langle a parbox with the table's caption\rangle}}

\begin{verbatim}
def\LT@caption{% \noalign{\egroup\LT@c@ption\@firstofone\LT@capti@n}
\end{verbatim}

\texttt{\textbackslash LT@c@ption} Caption command (with \texttt{optional argument}). \texttt{\protect} added in Version 3. \texttt{\fnum@table} added at V3.05.
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 package.

Actually this is not so bad, with FM leading the way.

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.
360 \@makecol
361 \@outputpage
362 \setbox\z@\vbox{\box\LT@head}%
End of \ifdim\dimen<\ht@\cclc.
363 \fi
End of \ifdim \ht\LT@lastfoot > \ht\LT@foot.
364 \fi
Reset \@colroom.
365 \global\@colroom@colht
366 \global\vsize@colht
Put the last page of the table on to the main vertical list.
367 \vbox
368 {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
End of \ifnum\outputpenalty > -\LT@end@pen.
369 \fi
Else \outputpenalty > -\@Mi.
370 \else
If we have not reached the end of the table,
371 \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
372 \@makecol
373 \@outputpage
Reset \vsize.
374 \global\vsize@colroom
Put the head at the top of the next page.
375 \copy\LT@head\nobreak
End of \ifnum\outputpenalty <-\@Mi.
376 \fi}

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.
377 \def\LT@end@hd@ft#1{%
378 \LT@echunk
Changed from \relax to \egraph at V3.04, see \LT@start.
379 \iffx\LT@start\egraph
380 \LT@err
381 \{Longtable head or foot not at start of table\%
382 \{Increase \LTchunksize\%
383 \fi
384 \setbox#1\box\z@
385 \LT@get@widths
386 \LT@chunk}

\endfirsthead Call \LT@end@hd@ft with the appropriate box.
\endfirsthead \def\endfirsthead{\LT@end@hd@ft\LT@firsthead}
\endfoot \def\endhead{\LT@end@hd@ft\LT@head}
\endlastfoot \def\endfoot{\LT@end@hd@ft\LT@foot}
\endlastfoot \def\endlastfoot{\LT@end@hd@ft\LT@lastfoot}
9.13 The \texttt{\textbackslash multicol} command

Earlier versions needed a special ‘draft’ form of \texttt{\textbackslash multicol}. This is not needed in version 4, and so these commands have been removed.

\texttt{\textbackslash LTmulticol}\texttt{\textbackslash LT@mwarn}

9.14 Footnotes

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

\texttt{\textbackslash LT@startpbox} Add extra code to switch the definition of \texttt{\@footnotetext}.

\begin{verbatim}
391 \def\LT@startpbox#1{\% 392 \bgroup 393 \let\@footnotetext\LT@p@ftntext 394 \setlength\hsize{#1}\% 395 \@arrayparboxrestore 396 \vrule \@height \ht\@arstrutbox \@width \z@}\% 397 \LT@endpbox 398 \@finalstrut\@arstrutbox 399 \egroup 400 \the\LT@p@ftn 401 \global\LT@p@ftn{}\% 402 \hfil}\% \end{verbatim}

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

\begin{verbatim}
397 \def\LT@endpbox{\% 398 \@finalstrut\@arstrutbox 399 \egroup 400 \the\LT@p@ftn 401 \global\LT@p@ftn{}\% 402 \hfil}\% \end{verbatim}

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

\begin{verbatim}
403 \def\LT@p@ftntext#1{\% 404 \edef\tempa{\the\LT@p@ftn\noexpand\footnotetext{\the\c@footnote}}\% 405 \global\LT@p@ftn\expandafter{\@tempa\#1}}\%
406 (/package)\% \end{verbatim}

................. longtable.sty ......................