

The ctable package*

Wybo Dekker
wybo@servalys.nl

April 30, 2008

Abstract

The `ctable` package provides a `ctable` command for the typesetting of table and figure floats. You will not need to type the usual nested `begin...end` sequences, as `ctable` is a command, not an environment. `ctable` has only 4 arguments, but the optional first one may hold many *key=value* pairs and makes `ctable` very flexible and extensible. It uses Simon Fear's `booktabs` package for better vertical spacing around horizontal rules and it provides facilities for making table footnotes.

1 Purpose

The `ctable` package lets you easily typeset captioned table and figure floats with optional footnotes. Both caption and footnotes will normally be forced within the width of the table. If the width of the table is specified, then `tabularx` will be used to typeset it, and one or more `X` column specifiers should be specified. Otherwise `tabular` will be used.

This package defines the commands `\ctable`, `\tnote` and `\tmark`, as well as four `\tabularnewline` generating commands. The latter generate reasonable amounts of whitespace around horizontal rules and are also useful for `tabulars` outside this package.

Since the `ctable` package imports the `array` and `booktabs` packages, all commands from those packages are available as well.

Note that, in line with the comments that Simon Fear made describing his `booktabs` package, vertical rules for column separation can be produced with `\ctable`, but no provisions are made to have them make contact with horizontal rules.

2 Usage

`\ctable` `\ctable` is called with 4 parameters, of which the first is optional:

```
\ctable[options]      % key=value,...
    {coldefs}          % for \begin{tabular}
    {foottable}        % zero or more \tnote commands (see below)
    {table rows}       % rows for the table
```

Options are given as *key=value* pairs, separated by comma's. Extra comma's, including one behind the last pair, don't hurt. Arguments to option should be put between braces if they contain comma's or equals signs. Currently the following option keys have been defined:

*This document corresponds to `ctable` v1.13, dated 2008/05/01.

<code>caption={...}</code>	table caption; the braces are needed only if your caption contains a comma or an equals sign.
<code>cap={...}</code>	for a short caption to go to the <code>\listoftables</code> . Without the <code>cap</code> option, the full caption will go into the <code>\listoftables</code> . If <code>cap</code> is given an empty value, no entry in the <code>\listoftables</code> will be made. This may be useful, for example, with the <code>continued</code> option.
<code>continued[=...]</code>	if used, the table will be numbered the same as the previous table. If used without an argument, the caption will be suffixed with ‘ (continued)’, if used with an argument, the suffix will be the argument.
<code>captionskip=...</code>	moves the caption relative to the table; the default is <code>0ex</code> , which puts captions at their default L ^A T _E X positions: a top caption’s baseline at <code>1ex</code> above the top rule position of the table and a bottom caption’s baseline at <code>4ex</code> below the bottom rule position.
<code>mincapwidth=...</code>	sets the minimum width of the float. Without this option, the width is set to that of the tabular, and the caption and footnotes are typeset within that width. This may be a problem with very narrow tables; <code>mincapwidth</code> can then be used to give the float a minimum width. The tabular will be centered in it.
<code>pos=...</code>	float position, default: <code>tbp</code> .
<code>label=...</code>	for <code>\label</code>
<code>width=...</code>	<code>tabularx</code> will be used to typeset the table at the specified width — one or more X column specifiers must be provided.
<code>maxwidth=...</code>	like the <code>width</code> option, but any X column specifiers will be replaced with <code>l</code> if the resulting table width would thus stay within the specified maximum width. This is especially useful where the L ^A T _E X source is generated by a script.
<code>center</code>	center the table in the available text width; this is the default.
<code>left</code>	left align the table in the available text width.
<code>right</code>	right align the table in the available text width.
<code>figure</code>	produce a figure float instead of a table float.
<code>botcap</code>	put the caption at the bottom of the float instead of on top of it.
<code>sideways</code>	rotate table or figure by 90 degrees anticlockwise and put it on a separate page. With the <code>twoside</code> option for the standard L ^A T _E X document classes, rotation will be -90 on even pages. If you use this option, the <code>pos</code> option is not allowed.
<code>star</code>	use the starred versions of the <code>table</code> and <code>figure</code> environments, which place the float over two columns when the <code>twocolumn</code> option or the <code>\twocolumn</code> command is active.
<code>nosuper</code>	in the footnote table, typeset footnote markers on the line, instead of superscripted.
<code>notespar</code>	typeset footnotes in a paragraph instead of in a table.
<code>framerule=...</code>	draw a frame around the table with the given rule thickness. The default is <code>0pt</code> , so that no frame will be seen.
<code>framesep=...</code>	set the distance between the frame and the table to the given dimension. The default is <code>0pt</code> .
<code>framefg=r g b</code>	set the foreground color of the frame (the rule color) to the given triplet of <i>rgb</i> -values. The values should be numbers between 0 and 1. The default is <code>0 0 0</code> (black).
<code>framebg=r g b</code>	set the background color of the frame (the color inside the frame) to the given triplet of <i>rgb</i> -values. The values should be numbers between 0 and 1. The default is <code>1 1 1</code> (white).

The footnotes are placed under the table, without a rule. You therefore probably will want to use the `\tnote` `\LL` (last line) command if you use footnotes. `\tnote[label]{footnote text}` places ^{label} footnote text

under the table. Can only be used in the footable parameter described above. The label is optional, the default label is a single a . For more detailed control, you can also replace this command with something like `labeltext&footnotetext\NN`.

`\tmark` `\tmark[label]` this command places the superscripted label in the table. It is equivalent with `$^{\label}$`. The label is optional, the default label is a single a .

The newline generating commands are a combination of `\tabularnewline` and zero or one of `booktabs` `\toprule`, `\midrule` or `\bottomrule`. These combinations have been made, and short names have been defined, because source texts for complex tables often become very crowded:

`\NN` Normal Newline, generates just a normal new line. An optional `dimen` parameter inserts extra vertical space under the line.

`\FL` First Line, generates a new line and a thick rule with some extra space under it. An optional `dimen` parameter sets the line width; the default is 0.08em.

`\ML` Middle Line: generates a new line and a thin rule with some extra space over and under it. An optional `dimen` parameter sets the line width; the default is 0.05em.

`\LL` Last Line: generates a new line and a thick rule with some extra space over it. An optional `dimen` parameter sets the line width; the default is 0.08em.

These macros can be used outside `\ctable` constructs.

Finally, for completeness, here are some of `booktabs`' commands that may be useful:

`\toprule` `\toprule[<wd>]` where `<wd>` is the optional thickness of the rule.

`\midrule` `\midrule[<wd>]`.

`\bottomrule` `\bottomrule[<wd>]`.

`\cmidrule` `\cmidrule[<wd>](<trim>){a-b}` where `<trim>` can be `r`, `l`, or `rl` and the rule is drawn over columns `a` through `b`.

`\morecmidrules` `\morecmidrules` must be used to separate two successive `cmidrules`.

`\addlinespace` `\addlinespace[<wd>]` inserts extra space between rows.

`\specialrule` `\specialrule{<wd>}{<abovespace>}{<belowspace>}`.

See the `booktabs` documentation for details.

2.1 The width and maxwidth options

When \LaTeX -sources containing tables are generated automatically by a script, it is often not known in advance what the maximum size of an `l` column will be. A good solution for this is to use an `X` specifier, typesetting the table at the text width with the `tabularx` package. However, this will result in too much white space in cases where the column contains small texts only. This problem can be solved by using the `maxwidth` option instead of the `width` option. The `X` specifiers will then be replaced with `l` as long as the width of the resulting table stays with the specified maximum width.

3 Examples

Table 1 is an example taken from the related package `threeparttable` by Donald Arseneau, with an extra footnote. It was typeset with:

```
\ctable[
  cap      = The Skewing Angles,
  caption  = The Skewing Angles ( $\beta$ ) for
             $\mu(H)+X_2$  and  $\mu(H)+HX$ ~\tmark,
  label    = nowidth,
]{rlcc}{
  \tnote{for the abstraction reaction,
     $\mu(H)+HX \rightarrow \mu(H)+X$ .}
  \tnote[b]{1 degree =  $\pi/180$  radians.}
  \tnote[c]{this is a particularly long note, showing that
```

Table 1: The Skewing Angles (β) for $\text{Mu}(\text{H}) + \text{X}_2$ and $\text{Mu}(\text{H}) + \text{HX}$ ^a

	$\text{H}(\text{Mu}) + \text{F}_2$	$\text{H}(\text{Mu}) + \text{Cl}_2$
$\beta(\text{H})$	80.9° ^b	83.2°
$\beta(\text{Mu})$	86.7°	87.7°

^a for the abstraction reaction,

$\text{Mu} + \text{HX} \rightarrow \text{MuH} + \text{X}$.

^b 1 degree = $\pi/180$ radians.

^c this is a particularly long note, showing that footnotes are set in raggedright mode as we don't like hyphenation in table footnotes.

```

footnotes are set in raggedright mode as we don't like
hyphenation in table footnotes.}
}{
&          & $\fam0 \text{H}(\text{Mu})+\text{F}_2$      & $\fam0 \text{H}(\text{Mu})+\text{Cl}_2$ \FL
&$\beta$(H) & $80.9^\circ\text{\tmark[b]}$ & $83.2^\circ$ \ML
&$\beta$(Mu) & $86.7^\circ$ & $87.7^\circ$ \NN
}

```

Table 2 is an example with a width specification, taken from the `tabularx` documentation, with the vertical rules removed. By using the trimming parameters of the `booktabs` `\cmidrule` command, some of the horizontal splitting was regained. The left option left aligns the table. It was typeset with:

```

\ctable[
  caption = Example with a specified width of 100mm,
  label   = width,
  width   = 100mm,
  pos     = h,
  left
]{c>\raggedright}Xc>\raggedright}X}{
  \tnote{footnotes are placed under the table}
}{
  \multicolumn{4}{c}{Example using tabularx} \FL
  \multicolumn{2}{c}{Multicolumn entry!} & THREE & FOUR \ML
  \cmidrule(r){1-2}\cmidrule(rl){3-3}\cmidrule(l){4-4} \NN
  one&
  The width of this column depends on the width of the
    table.\tmark &
  three&
  Column four will act in the same way as
  column two, with the same width. \LL
}

```

Table 2: Example with a specified width of 100mm

Example using tabularx			
	Multicolumn entry!	THREE	FOUR
one	The width of this column depends on the width of the table. ^a	three	Column four will act in the same way as column two, with the same width.

^a footnotes are placed under the table

Figures, even single ones, are always put in tabular cells. This is not particularly handy for single pictures, but it eases the construction of arrays of pictures, including sub-captions, delineation, and spacing. Figure 1 shows a figure that has been produced with the `\ctable` command, in combination with `\usepackage{carom}` it has been typeset with:

```
\ctable[
  figure,
  botcap,
  pos      = h,
  caption  = The di- and tri-bromobenzenes,
  label    = fig,
  framebg  = .53 .81 .92,
  framerule = 1pt,
  framesep = 4ex,
]{ccc}{\NN
  \bzdrv{1==Br;2==Br}&
  \bzdrv{1==Br;3==Br}&
  \bzdrv{1==Br;4==Br}
  1,2 & 1,3 & 1,4
  \NN[3ex]
  \bzdrv{1==Br;2==Br;3==Br}&
  \bzdrv{1==Br;2==Br;4==Br}&
  \bzdrv{1==Br;3==Br;5==Br}
  1,2,3 & 1,2,4 & 1,3,5
}
```

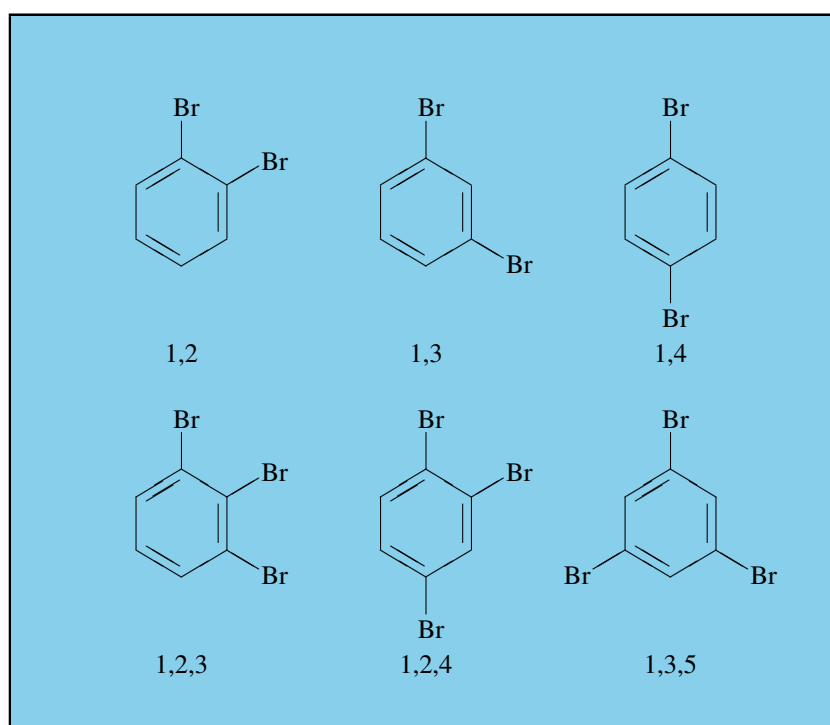


Figure 1: The di- and tri-bromobenzenes

(The excessive whitespace at the left of the figure is caused by the bounding boxes generated by the `carom` package.)

4 Option examples

In the following, small examples will be shown illustrating the effect of options. In the left column the relevant part of the source is shown, in the right column you see the result. In most cases you see

a standard example on a light yellow background, followed by one or more variations on a light blue background. Where necessary, the example will show boxes to indicate the page and the text body.

4.1 center, left, right

These options align the float in the page; the default is `center`:

```
\ctable[
  caption = Centered,
]{c}{\FL Table's first row\LL}
}
```

Table 1: Centered
Table's first row

```
\ctable[
  caption = Left,
  left
]{c}{\FL Table's first row\LL}
}
```

Table 1: Left
Table's first row

```
\ctable[
  caption = Right,
  right
]{c}{\FL Table's first row\LL}
}
```

Table 1: Right
Table's first row

4.2 nosuper

Footnote markers in `ctable` are typeset superscripted by default. Use the `nosuper` option to place them on the base line:

```
\ctable{c}{
  \tnote{First footnote}
  \tnote[b]{Second footnote}
}{\FL Table's\tnote\ first\tnote[b]\ row\LL}
}
```

Table's ^a first ^b row

^a First footnote

^b Second footnote

```

\cetable[nosuper]{c}{
  \tnote[a.]{First footnote}
  \tnote[b.]{Second footnote}
}{\FL Table's\tmark\ first\tmark[b]\ row\LL}

}

```

Table's^a first^b row

a. First footnote

b. Second footnote

4.3 notespar

Footnotes in ctable are typeset in a paragraph, instead of a table:

```

\cetable{c}{
  \tnote{First note}
  \tnote[b]{Second note}
  \tnote[c]{Third note}
}{\FL Table's\tmark\ first\tmark[b]\ row
  with footnotes\tmark[c]\LL}

}

```

Table's^a first^b row with footnotes^c

^a First note

^b Second note

^c Third note

```

\cetable[notespar]{c}{
  \tnote[a]{First note.}
  \tnote[b]{Second note.}
  \tnote[c]{Third note.}
}{\FL Table's\tmark\ first\tmark[b]\ row
  with footnotes\tmark[c]\LL}

}

```

Table's^a first^b row with footnotes^c

^a First note. ^b Second note. ^c Third note.

4.4 continued

The continued option suffixes the caption with ' (continued)', and lowers the table number by one, so that it obtains the same number as the previous table. This option can be given an argument to replace the default suffix:

```

\cetable[
  caption = Caption,
  mincapwidth = 50mm,
]{c}{\FL Table's first row\LL}

}

```

Table 1: Caption

Table's first row

```

\cetable[
  caption = Caption,
  mincapwidth = 50mm,
  continued
]{c}{\FL Table's first row\LL}

}

```

Table 1: Caption (continued)

Table's first row

```

\cetable[
  caption = Caption,
  mincapwidth = 50mm,
  continued = \textit{(contd)}
]{c}{\FL Table's first row\LL}

}

```

Table 1: Caption (*contd*)

Table's first row

4.5 mincapwidth

`ctable` forces caption and footnotes to stay within the width of the table. Sometimes, however, tables are so narrow, that this is not really what you want. In such cases, use the `mincapwidth` option to give caption and footnotes some extra room:

```
\ctable[
  caption = a lengthy caption
]{c}{\FL row1\LL}
}
```

Table 1: a
lengthy
caption

row1

```
\ctable[
mincapwidth = 55mm,
  caption = a lengthy caption
]{c}{\FL row1\LL}
}
```

Table 1: a lengthy caption

row1

You can set `mincapwidth` to a large value, say `\hsize`, if you want a one-line caption. Note, however, that this may influence the horizontal positioning of the table: values larger than `\hsize` will move a centered table out of the center, a value of `\hsize` will prevent the `left` and `right` options to do their work, because the table is already captured between the left and right margins.

4.6 maxwidth

When \LaTeX -sources containing tables are generated automatically by a script, it is often not known in advance what the maximum size of an l column will be. A good solution for this is to use an X specifier, typesetting the table at the text width with the `tabularx` package. However, this will result in too much white space in cases where the column contains small texts only. This problem can be solved by using the `maxwidth` option instead of the `width` option. The X specifiers will then be replaced with l as long as the width of the resulting table stays with the specified maximum width.

```
\ctable[
  framerule = .1pt,
  maxwidth=3cm
]{lX}{\FL 1 & first row\LL}
}
```

1	first row
---	-----------

```
\ctable[
  framerule = .1pt,
  maxwidth=3cm
]{lX}{\FL 1 & test\LL}
}
```

1	test
---	------

4.7 framerule

The following examples show the use of frames and backgrounds. Every table is typeset by `ctable` with a frame around it, but the frame is, by default, drawn with a zero width line, and is therefore invisible. You can make it visible by either changing the linewidth to a positive value or by giving it a background color, which will be used to fill the frame.

Here is a simple table without a frame, followed by one with a red, 1pt thick frame:

```
\ctable[
  caption = Frame,
]{c}{c}{\FL Table's first row\LL}
}
```

Table 1: Frame

Table's first row

```
\ctable[
  caption = Frame,
  framerule = 2pt,
  framefg = .8 0 0
]{c}{c}{\FL Table's first row\LL}
}
```

Table 1: Frame

Table's first row

As you see, the frame fits closely to the first (`\FL`) and last (`\LL`) table lines. This can be a reason to either remove those lines, or to introduce some whitespace between the frame and the table with the `framesep` option:

```
\ctable[
  caption = Frame,
  framerule = 1pt,
  framefg = .8 0 0,
  framesep=10pt
]{c}{c}{\FL Table's first row\LL}
}
```

Table 1: Frame

Table's first row

And finally, we could also frame the table by giving it a, say, yellow background instead of a red frame line, or even do both:

```
\ctable[
  caption = Frame,
  framebg = 1 1 0,
  framesep=10pt
]{c}{c}{\FL Table's first row\LL}
}
```

Table 1: Frame

Table's first row

```
\ctable[
  caption = Frame,
  framerule = 2pt,
  framesep = 5pt,
  framebg = 1 1 0,
  framefg = 1 0 0,
  framesep=10pt
]{c}{c}{\FL Table's first row\LL}
}
```

Table 1: Frame

Table's first row

4.8 captionskip

The distance between a top caption and the table is 2ex, but it can be varied with `captionskip`:

```
\ctable[
  caption = Caption,
]{c}{\FL Table's first row\LL}
}
```

Table 1: Caption

Table's first row

```
\ctable[
  caption = Caption,
  captionskip = 1ex,
]{c}{\FL Table's first row\LL}
}
```

Table 1: Caption

Table's first row

This works for bottom caption, too:

```
\ctable[
  caption = Caption,
  botcap
]{c}{\FL Table's first row\LL}
}
```

Table's first row

Table 1: Caption

```
\ctable[
  caption = Caption,
  captionskip = -2ex,
  botcap
]{c}{\FL Table's first row\LL}
}
```

Table's first row

Table 1: Caption

4.9 figure, botcap

By default, `ctable` generates a table float, but with the `figure` option, a figure float is generated instead. The caption stays on top, so if you are accustomed to have bottom caption for your figures, you will probably also need the `botcap` option:

```
\ctable[caption = a table]{c}{
}{\FL Table's first row\LL}
}
```

Table 1: a table

Table's first row

```
\let\H\hsize
\ctable[
  caption = a figure,
  figure, botcap,
  width=.4\H,
]{@{}>{\H=.4\H}X>{\H=.6\H}X@{}~}{\FL
\includegraphics[width=\H]{penguin}&
\includegraphics[width=\H]{lion}\LL
}
}
```

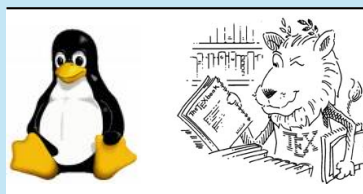


Figure 1: a figure

4.10 sideways

The `sideways` option creates a landscape table with its head pointing at the spine — when the document-class’ `twoside` option has been used, that is. The following examples show the effect of the `sideways` option, first on page one, then on page 2. Note that the `caption` option has not been used, so no caption appears:

```
\ctable[
]{c}{ }\{\FL first row\LL\}
}
```

first row

```
\ctable[
  sideways
]{c}{ }\{\FL first row\LL\}
}
```

first row

```
\setcounter{page}{2}
\ctable[
  sideways
]{c}{ }\{\FL first row\LL\}
}
```

first row

5 Implementation

```

1 \RequirePackage{color,xkeyval,array,tabularx,booktabs,rotating,caption}
2 \def\NN{\tabularnewline}
3 \def\FL{\toprule}
4 \def\ML{\NN\midrule}
5 \def\LL{\NN\bottomrule}
6 \def\@ctblfgcolor#1 #2 #3={\definecolor{@ctblframefg}{rgb}{#1,#2,#3}}
7 \def\@ctblbgcolor#1 #2 #3={\definecolor{@ctblframebg}{rgb}{#1,#2,#3}}
8 \let\@cttextsuperscript\@textsuperscript
9 \def\@ctcopy#1{\footnotesize #1}
10 \newdimen\@ctblframesep
11 \newdimen\@ctblframerule
12 \newdimen\@ctblwidth
13 \newdimen\@ctblcaptionskip
14 \newdimen\@ctblmaxwidth
15 \newdimen\@ctblmincapwidth
16 \newdimen\@ctblw % the final width
17 \newdimen\@ctblfloatwidth
18 \newdimen\@ctbloldsep
19 \newdimen\@ctbloldrule

```

Allocate box registers so that we can determine the widths of the tables

```
20 \newbox\ctbl@t % tabular saved and measured here
```

Option setting commands from keyval. The table position (here, top, bottom, page) gets a special treatment, since \LaTeX does not expand commands there. So instead of putting things like `tbp` in a command like `\@ctblbegin` we put `\begin{table}[tbp]` in it.

```

21 \define@key{ctbl}{caption}{\def\@ctblcaption{#1}}
22 \define@key{ctbl}{cap}{\def\@ctblcap{#1}}
23 \define@key{ctbl}{label}{\def\@ctbllabel{#1}}
24 \define@key{ctbl}{continued}{(continued)}{\def\@ctblcontinued{#1}}
25 \define@key{ctbl}{pos}{\def\@ctblpos{#1}\def\@ctblbegin{\@ctblbeg[#1]}}
26 \define@key{ctbl}{width}{\@ctblwidth=#1}
27 \define@key{ctbl}{maxwidth}{\@ctblmaxwidth=#1}
28 \define@key{ctbl}{mincapwidth}{\@ctblmincapwidth=#1}
29 \define@key{ctbl}{botcap}{\def\@ctblbotcap{1}}
30 \define@key{ctbl}{sideways}{\def\@ctblsideways{sideways}}
31 \define@key{ctbl}{rotate}{\def\@ctblsideways{sideways}}
32 \PackageWarning{ctable}{
33     using obsolete option 'rotate', use 'sideways' instead}
34 }
35 \define@key{ctbl}{figure}{\def\@ctbltaborfig{figure}}
36 \define@key{ctbl}{center}{\def\@ctblalign{center}}
37 \define@key{ctbl}{right}{\def\@ctblalign{flushright}}
38 \define@key{ctbl}{left}{\def\@ctblalign{flushleft}}
39 \define@key{ctbl}{star}{\def\@ctblstarred{*}}
40 \define@key{ctbl}{framerule}{\@ctblframerule=#1}
41 \define@key{ctbl}{framesep}{\@ctblframesep=#1}
42 \define@key{ctbl}{framefg}{\@ctblfgcolor#1=}
43 \define@key{ctbl}{framebg}{\@ctblbgcolor#1=}
44 \define@key{ctbl}{captionskip}{\@ctblcaptionskip=#1}
45 \define@key{ctbl}{nosuper}{\let\@cttextsuperscript\@ctcopy}
46 \define@key{ctbl}{notespar}{\def\@ctblnotespar{1}}

```

A caption will only be generated if the *caption* option was used. First adapt `\@caption` so that it does not insert empty (short) captions in the *lot/lof*

```

47 \long\def\@caption#1[#2]#3{%
48   \par
49   \ifx#2\empty\else\addcontentsline{\csname ext@#1\endcsname}{#1}%
50     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}\fi%
51   \begingroup
52     \@parboxrestore
53     \if@minipage
54       \@setminipage

```

```

55 \fi
56 \normalsize
57 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
58 \endgroup}
59 \def\@ctblCaption{
60 \ifx\@ctblcap\undefined\let\@ctblcap\@ctblcaption\fi
61 \ifx\@ctblcaption\empty\else
62 \caption[\@ctblcap]{\label{\@ctbllabel}\@ctblcaption\ \@ctblcontinued}
63 \fi
64 }
65 \def\@ctblframe#1#2#3{%
66 \@ctbloldsep\fbboxsep\fbboxsep\@ctblframesep%
67 \@ctbloldrule\fbboxrule\fbboxrule\@ctblframerule%
68 \fcolorbox{#1}{#2}{\fbboxsep\@ctbloldsep\fbboxrule\@ctbloldrule #3}%
69 }
70 \newcommand{\tnote}[2][a]{%
71 \ifx\@ctblnotespar\empty%
72 \hbox{\@cttextsuperscript{\normalfont\textit{#1}}}&#2\NN
73 \else%
74 \@cttextsuperscript{\normalfont\textit{#1}}\,,#2
75 \fi
76 }
77 \newcommand{\tmark}[1][a]{%
78 \hbox{\textsuperscript{\normalfont\textit{#1}}}}
79 \newcommand{\ctable}[4][{}]{%
80 \def\@ctbltaborfig{table}%
81 \def\@ctblalign {center}%
82 \def\@ctblsideways{}%
83 \def\@ctblcontinued{}%
84 \def\@ctblpos {}%
85 \def\@ctblcaption {}%
86 \let\@ctblcap\undefined%
87 \def\@ctbllabel {}%
88 \def\@ctblbeg {\begin{\@ctblsideways\@ctbltaborfig\@ctblstarred}}%
89 \def\@ctblbegin {\@ctblbeg}%
90 \def\@ctblend {\end{\@ctblsideways\@ctbltaborfig\@ctblstarred}}%
91 \def\@ctblbotcap {}%
92 \def\@ctblstarred {}%
93 \def\@ctblnotespar{}%
94 \definecolor{\@ctblframefg}{rgb}{0,0,0}%
95 \definecolor{\@ctblframebg}{rgb}{1,1,1}%
96 \@ctblframerule0pt
97 \@ctblcaptionskip0ex
98 \@ctblframesep0pt
99 \@ctblwidth=0pt
100 \@ctblmaxwidth=0pt
101 \@ctblmincapwidth=0pt
102 \setkeys{ctbl}{#1}%

It makes no sense to use width together with maxwidth or pos together with sideways
103 \ifdim\@ctblwidth=0pt\else
104 \ifdim\@ctblmaxwidth=0pt\else
105 \PackageError{ctable}{
106 You may not use the width and maxwidth options together}{
107 Use either width or maxwidth}
108 \fi
109 \fi
110 \ifx\@ctblpos\empty\else
111 \ifx\@ctblsideways\empty\else
112 \PackageError{ctable}{
113 You may not use the pos and sideways options together}{
114 Rotated tables and figures are always typeset on a separate page}
115 \fi
116 \fi

```

It makes no sense to label a captionless table, because the label can't be placed, leaving the user wondering why references to the table get a ??

```

117 \ifx\@ctblcaption\empty
118 \ifx\@ctbllabel\empty\else
119 \PackageError{ctable}{
120     You may not label a captionless table}{
121     Such a label can't be referenced}
122 \fi
123 \fi
124 \newcolumnntype{Y}{X}% save the X type

```

save the table contents in a box, so we can determine its width, initially, save the table typeset with the tabular environment:

```

125 \sbox\ctbl@t{%
126     \newcolumnntype{X}{l}% temporarily make type X = l
127     \@ctblframe{@ctblframefg}{@ctblframebg}{%
128         \begin{tabular}{#2}
129         #4%
130         \end{tabular}%
131     }%
132 }%

```

then look if we'll need the tabularx environment:

```

133 \newif\if@ctblusex\@ctblusexfalse
134 \ifdim\@ctblmaxwidth=0pt
135 \ifdim\@ctblwidth=0pt
136 \else
137 \@ctblusextrue
138 \fi
139 \else
140 \ifdim\wd\ctbl@t>\@ctblmaxwidth
141 \@ctblusextrue
142 \fi
143 \fi
144 %
145 % if so, replace tabular with tabularx:
146 %
147 \if@ctblusex
148 \sbox\ctbl@t{%
149     \newcolumnntype{X}{Y}% restore X
150     \@ctblframe{@ctblframefg}{@ctblframebg}{%
151         \begin{tabularx}{\ifdim\@ctblwidth>0pt\@ctblwidth\else\@ctblmaxwidth\fi}{#2}
152         #4%
153         \end{tabularx}%
154     }%
155 }%
156 \fi

```

the `ctbl@t` box now contains the table as we want to typeset it; determine its width:

```

157 \@ctblw=\wd\ctbl@t

```

Now find the width of the float, `\@ctblfloatwidth`; everything in it will be centered within that width. Normally we'll use the width of the table, `\@ctblw`, but if the mincapwidth, `\@ctblmincapwidth` was set wider than the table, that will be used:

```

158 \@ctblfloatwidth=\ifdim\@ctblmincapwidth>\@ctblw
159 \@ctblmincapwidth
160 \else
161 \@ctblw
162 \fi

```

`\@ctblbegin` is now defined as something like `\begin{table}[tbp]`.

```

163 \@ctblbegin
164 \ifx\@ctblcontinued\empty\else\ContinuedFloat\fi
165 \begin{@ctblalign}
166     \begin{minipage}{\@ctblfloatwidth}\parindent0pt
167     \ifx\@ctblbotcap\empty\@ctblCaption\vskip\@ctblcaptionskip\fi
168     \centering{\usebox\ctbl@t} % insert the tabular

```

```

169      \def\@ctblfootnotes{#3}%
170      \ifx#3\empty\else % append footnotes, if any
171        \ifx\@ctblnotespar\empty%
172          \\
173          \begin{tabularx}{\hspace}{r@{\,}>{\footnotesize\raggedright}X}
174            #3%
175          \end{tabularx}%
176        \else%
177          \[.2ex]
178          \begin{minipage}{\hspace}\footnotesize\raggedright%
179            #3%
180          \end{minipage}%
181        \fi
182      \fi
183      \ifx\@ctblbotcap\empty\else\vskip\@ctblcaptionskip\@ctblCaption\fi
184      \end{minipage}
185    \end{\@ctblalign}
186  \@ctblend
187 }

```

Change History

v1.00	General: First release.	1	v1.08	General: Standardized file setup following http://www.ctan.org/tex-archive/info/dtxtut/dtxtut.pdf mincapwidth option added Moved newdi-men definition outside ctable macro	1
v1.01	General: Making use of booktabs package	1			
v1.02	General: Using keyval to reduce args to 4	1	v1.09	General: Added option nosuper; corrected in-correct positioning when table is wider than mincapwidth	1
v1.03	General: Many syntactic corrections, thanks to Johannes Braams	1	v1.10	General: Footnote markers now stay superscript with nosuper. Documentation: added many examples for the options. Corrected some unwanted white space in captions. Caption package included to correct booktabs errors in caption position. And for later use of its facilities. *Captionskip option redefined*: Opt value now corresponds to LaTeX defaults	1
v1.04	General: Caption, if empty, will not be typeset rotate option added star option added to use table* and figure* environments	1	v1.11	General: Added some percent signs at EOL to prevent whitespace, Removed xspace usage - caused overfull badness	1
v1.05	General: maxwidth option added	1	v1.12	General: Option notespar added	1
v1.06	General: left, right and center options added framesep,rule,fg,bg options added error in width-setting corrected	1	v1.13	General: cap option with empty argument will not be inserted in lot/lof Added option con-tinued, for continuation tables: same num-ber as previous table, ‘(continued)’ added to caption.	1
v1.06a	General: two errors corrected: made setting fboxsep and fboxrule only temporary re-moved superfluous space after tabulars . . .	1			
v1.06b	General: Added several at eol to remove super-fluous whitespace occurring sometimes . . .	1			
v1.07	General: Added option sideways, option rotate now obsolete; added option captionskip . . .	1			