The ctable package*

Wybo Dekker
wybo@dekkerdocumenten.nl

October 18, 2015

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.

Contents

1 Introduction

2 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

---

[*] This document corresponds to ctable v1.31, dated 2015/10/17.

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`, and 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.

3 Usage

`\setupctable` `\ctable` defaults can be set, either in the preamble or in the body, with:

```
\setupctable{options}  % key=value,...
```

`\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.

4 Options

Currently the following option keys have been defined:

`bgopacity=...`	Sets the opacity of the table's background color, where 1 is 100% opaque (the default), and 0 is completely transparent. One application is with watermarking: most watermarking packages print their watermark on the background. `ctable`'s background color, which is opaque by default, may make the watermark (partially) invisible. You can avoid this by setting the `bgopacity` option to a value lower than 1. Note that this works only in PDF mode, a warning is issued otherwise. Note: there are two limitations to transparency setting: 1. it works only in PDF mode: so it works in pdflatex and lualatex, but is disabled in xelatex. 2. it disables transparency features in the tikz package; therefore, ctable checks if the tikz package is loaded and if so, disables its own transparency with a warning. That helps only if you load tikz *before* ctable.
`botcap`	put the caption at the bottom of the float instead of on top of it. See also: `topcap`, `sidecap`.
`caption=...`	table caption; the braces are needed only if your caption contains a comma or an equals sign.
`cap=...`	for a short caption to go to the `\listoftables`. Without the `cap` option, the full caption will go into the `\listoftables`. If `cap` is given an empty value, *and you have loaded the `caption` package*, no entry in the `\listoftables` will be made. This may be useful, for example, with the `continued` option.
`captionskip=...`	moves the caption relative to the table; the default is `0ex`, which puts captions at their default LaTeX positions. For the standard LaTeX classes this means that a top caption's baseline at `1ex` above the top rule position of the table and a bottom caption's baseline at `4ex` below the bottom rule position. These dimensions may be different for other classes or when other packages are included. The `memoir` class and the `caption` package, for example, both typeset captions differently, and the

combination of both even differs from each alone.[1] Keep in mind that when you use the `caption` package in the `memoir` class, `memoir`'s caption commands are suspended and `caption`'s commands must be used.

`captionsleft`	This option is defined for `\setupctable` only, and it is effective only where the `sideways` option is used. After `\setupctable{captionsleft}` all tables typeset with the `sideways` option will have their captions at the left.
`captionsright`	This option is defined for `\setupctable` only, and it is effective only where the `sideways` option is used. After `\setupctable{captionsright}` all tables typeset with the `sideways` option will have their captions at the right.
`captionsinside`	This option is defined for `\setupctable` only, it is the default, and it is effective only where the `sideways` option is used. After `\setupctable{captionsinside}` all tables typeset with the `sideways` option will have their captions at the left in one-sided documents. In twosided documents, captions will be on the left for odd-numbered pages and on the right for even-numbered pages. This is the default.
`center`	center the table in the available text width; this is the default. See also: `left`, `right`.
`continued=...`	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.
`doinside=...`	command to be run inside, just before the tabular or tabularx environment. You can use this, for example, for the adjustment of the font size with `\small`.
`figure`	produce a figure float instead of a table float. See also: `table`.
`footerwidth=...`	Footnotes are typeset within the width of the table. When you use the `mincapwidth` option, presumably because the table is very narrow, footnotes are given the same width as the caption. With small footnotes this may not be what you want; this option can be used to give the footnotes their own width. Without an argument, they will be typeset within the width of of the table.
`framebg=`*r g b*	set the background color of the frame (the color inside the frame) to the given triplet of *rgb*-values. The values should be numbers between 0 and 1. The default is `1 1 1` (white).
`framefg=`*r g b*	set the foreground color of the frame (the rule color) to the given triplet of *rgb*-values. The values should be numbers between 0 and 1. The default is `0 0 0` (black).
`framerule=...`	draw a frame around the table with the given rule thickness. The default is `0pt`, so that no frame will be seen.
`framesep=...`	set the distance between the frame and the table to the given dimension. The default is `0pt`.
`label=...`	labels the float with `\label`.
`left`	left align the table in the available text width. See also: `center`, `right`.
`maxwidth=...`	like the *width* option, but any `X` column specifiers will be replaced with `l` if the resulting table width would thus stay within the specified maximum width. This is especially useful where the LaTeX source is generated by a script.
`mincapwidth=...`	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; `mincapwidth` can then be used to give the float a minimum width. The tabular will be centered in it. If you don't want the footnotes to be affected see the `footerwidth` option.
`nonotespar`	typeset footnotes in a table; this is the default. See also: `notespar`.
`nosideways`	undo the sideways option. See also: `sideways`.
`nostar`	use the un-starred versions of the `table` and `figure` environments; this is the default
`nosuper`	in the footnote table, typeset footnote markers on the line, instead of superscripted.
`notespar`	typeset footnotes in a paragraph instead of in a table.
`pos=...`	float position, default: `tbp`.
`right`	right align the table in the available text width.
`sidecap`	put the caption at the side of the float. Currently, this works only if you have loaded the `memoir`

---

[1] I did some measurements on the whitespace between the caption and the top of the table with and without using the caption package and/or the memoir class: standard LaTeX: 1ex; memoir: 2.32ex; caption: 2.69ex; both memoir and caption: 2.68ex. For the distances between bottom caption baselines and the table bottom I found, respectively: 3.90ex, 3.41ex, 3.72ex and 3.74ex

	class, otherwise an error message is generated. The parameters for the caption, such as its vertical positioning, width and more, must be set with the appropriate `memoir` commands. See also: `botcap`, `topcap`.
`sideways`	rotate table or figure by 90 degrees anticlockwise and put it on a separate page. With the twoside option for the standard LaTeX document classes, rotation will be -90 on even pages, unless the options `captionleft` or `captionsright` are used. If you use this option, the `pos` option is not allowed. See also: `nosideways`, `captionsinside`.
`star`	use the starred versions of the `table` and `figure` environments, which place the float over two columns when the `twocolumn` option or the `\twocolumn` command is active. See also: `nostar`.
`super`	in the footnote table, typeset footnote markers as superscripts; this is the default. See also: `nosuper`.
`table`	produce a table float (this is the default). See also: `figure`.
`topcap`	put the caption top of the float; this is the default. See also: `botcap`, `sidecap`.
`width=...`	`tabularx` will be used to typeset the table at the specified width — one or more `X` column specifiers must be provided.

5 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.

6 Tables wider than the text width

When you make a table wider than `\textwidth`, it will extend in the right margin. If it is a large table, occupying a whole page, you can use the geometry package and surround your ctable call with `\newgeometry{width=...,margin=...}` and `\restoregeometry`. However, both geometry commands imply `\clearpage`, so your table will appear on an otherwise empty page.

Alternatively, you can center the table on the paper, extending in both margins, by using the option `doinside=\hspace*{<dimen>}` with an appropriate negative `dimen>`.

7 Setting option defaults: setupctable

`\setupctable`

Every call of `\ctable` resets the options to their defaults before evaluating the first (optional) argument. So if you make two ctables: `\ctable[left,...` and `\ctable[...`, the first will be left-aligned on the page, but the second, lacking the `left` option, will be centered, because that is the default. If you want all your tables left-aligned, it's more practical to change the default by calling `\setupctable{left}`, either in the preamble or somewhere in the body. In latter case only tables following the call will have their defaults changed.

`\setupctable` can set the defaults for all options except (of course) `caption`, `cap`, and `label`. Actually, the initial option defaults are set by calling `\setupctable` as follows:

```
\setupctable{
  captionskip=0pt,        framerule=0pt,    nostar,
  center,                 framesep=0pt,     pos=tbp,
  continued=(continued),  maxwidth=0pt,     super,
  doinside={},            mincapwidth=0pt,  table,
  framebg=1 1 1,          nonotespar,       topcap,
  framefg=0 0 0,          nosideways,       width=0pt
}
```

8 Other commands

`\tnote` `\tnote[label]{footnote text}` places ^{label} footnote text under the table. This command can only be used in `\ctable`'s third argument, i.e. the foottable argument 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`. The footnotes are placed under the table, without a rule. You therefore probably will want to use the `\LL` (last line) command if you use footnotes.

`\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*. `\tmark` may be used in captions, but only *without* an argument.

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. Is an alias for `\tabularnewline`

`\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. Is an alias for `\toprule`

`\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. Is an alias for `\tabularnewline\midrule`

`\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. Is an alias for `\tabularnewline\bottomrule`

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 thinkness 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.

9 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
             $\fam0 Mu(H)+X_2$ and $\fam0 Mu(H)+HX$~\tmark,
   label   = nowidth,
   pos     = h
]{rlcc}{
   \tnote{for the abstraction reaction,
          $\fam0 Mu+HX \rightarrow MuH+X$.}
   \tnote[b]{1 degree${} = \pi/180$ radians.}
   \tnote[c]{this is a particularly long note, showing that
             footnotes are set in raggedright mode as we don’t like
             hyphenation in table footnotes.}
}{                                                          \FL
  &            & $\fam0 H(Mu)+F_2$     & $\fam0 H(Mu)+Cl_2$ \ML
  &$\beta$(H)  & $80.9^\circ$\tmark[b] & $83.2^\circ$       \NN
  &$\beta$(Mu) & $86.7^\circ$          & $87.7^\circ$       \LL
}
```

Table 1: The Skewing Angles (β) for Mu(H) + X_2 and Mu(H) + HX [a]

		H(Mu) + F_2	H(Mu) + Cl_2
	β(H)	80.9°[b]	83.2°
	β(Mu)	86.7°	87.7°

[a] for the abstraction reaction, Mu + HX → MuH + 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.

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     = ht,
   left
]{c>{\raggedright}Xc>{\raggedright}X}{
   \tnote{footnotes are placed under the table}
}{                                                         \FL
   \multicolumn{4}{c}{Example using tabularx}              \ML
   \multicolumn{2}{c}{Multicolumn entry!} & THREE & FOUR   \NN
       \cmidrule(r){1-2}\cmidrule(rl){3-3}\cmidrule(l){4-4}
   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
}
```

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

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

spacing. For a small example, which also shows how you can simplify the construction of figure arrays, see subsection 10.9 on page 11.

10 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.

10.1 `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}
```

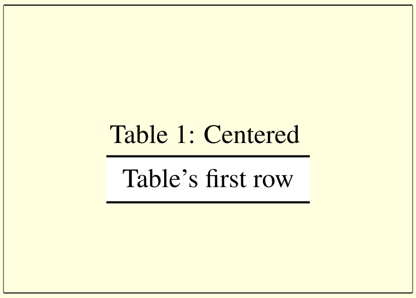


Table 1: Centered

Table's first row

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

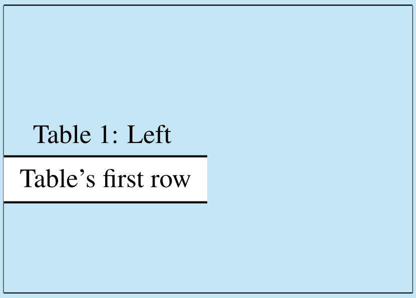


Table 1: Left

Table's first row

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

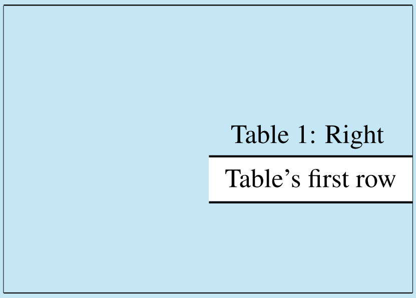


Table 1: Right

Table's first row

10.2 `super, 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\tmark\ first\tmark[b]\ row\LL}
```

Table's[a] first[b] row

[a] First footnote
[b] Second footnote

```
\ctable[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

10.3 `notespar, nonotespar`

By default, footnotes in `ctable` are typeset in a table, one line per note. This corresponds with the `nonotespar` option.You can also typeset them in a paragraph, one after the other, by using the `notespar` option:

```
\ctable{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

```
\ctable[notespar]{c}{
  \tnote[a]{First note.}
  \tnote[b]{Second note.}
  \tnote[c]{Third note, this one is a
            little longer and forces a
            new line at the end.\\}
  \tnote[d]{And here is e very long note:
            \input{thuan}}
}{\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, this one is a little longer and forces a new line at the end.
[d] And here is e very long note: Had our solar system included two suns, the problem would have involved three bodies (the two suns and each planet), and chaos would have been immediately obvious. Planets would have had erratic and unpredictable orbits, and creatures living on one of these planets would never have been able to percieve the slightest harmony. Nor would it have occurred to them that the universe might be ruled by laws and that it is up to man's intellect to discover them. Besides, it is not at all obvious that life and conscience could even emerge in such a chaotic system.

10.4 `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:

```
\ctable[
  caption = Caption,
  mincapwidth = 50mm,
]{c}{}{\FL Table’s first row\LL}
```

Table 1: Caption

Table's first row

```
\addtocounter{table}{1} % remove for source
\ctable[
  caption = Caption,
  mincapwidth = 50mm,
  continued
]{c}{}{\FL Table's first row\LL}
```

Table 1: Caption (continued)

Table's first row

```
\addtocounter{table}{1} %remove for source
\ctable[
  caption = Caption,
  mincapwidth = 50mm,
  continued = \textit{(contd)}
]{c}{}{\FL Table's first row\LL}
```

Table 1: Caption *(contd)*

Table's first row

10.5 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 = 40mm,
  caption = a lengthy caption
]{c}{\tnote{this is a footnote}}
{\FL row1\tmark\LL}
```

Table 1: a lengthy caption

row1^a

^{a} this is a footnote

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. When footnotes are small, you may wish to undo the effect of the `mincapwidth` option on them:

```
\ctable[
  mincapwidth = 40mm,
  footerwidth,
  caption = a lengthy caption
]{c}{\tnote{footnote}}
{\FL row1\tmark\LL}
```

Table 1: a lengthy caption

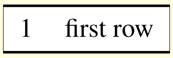
row1^a

^{a} footnote

10.6 maxwidth

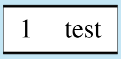
When LaTeX-sources containing tables are generated automatically by a script, it is often not known in advance what the maximum size of an `l` column will be. A good solution for this is to use an `X` specifier, typesetting the table at the text width with the `tabularx` package. However, this will result 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}
```



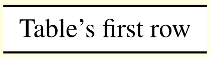
1	test

10.7 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}{}{\FL Table’s first row\LL}
```

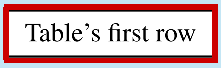
Table 1: Frame



Table’s first row

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

Table 1: Frame

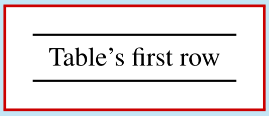


Table’s first row

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}{}{\FL Table’s first row\LL}
```

Table 1: Frame

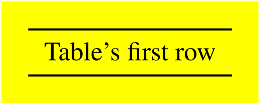


Table’s first row

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

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

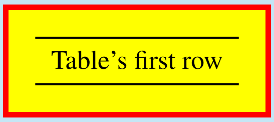
Table 1: Frame



Table’s first row

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

Table 1: Frame



Table’s first row

10.8 `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

10.9 `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

```
\newcommand{\F}[1]{
  \includegraphics[width=\hsize]{#1}
}
\newcolumntype{H}[1]{>{\hsize=#1\hsize}X}
\ctable[
  caption = a figure,
  figure, botcap,
  width=.4\hsize,
]{H{.4}H{.6}}{}{\FL
  \F{penguin}& \F{lion}\LL
}
```

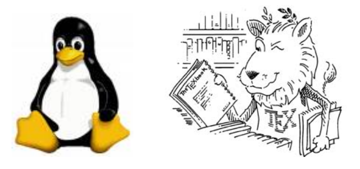


Figure 1: a figure

10.10 `doinside`

The argument of doinside is supposed to be a command to be run inside, just before the tabular or tabularx environment. You can use this, for example, for the adjustment of the font size with `\small`:

```
\ctable[
  caption=Doinside,
  doinside = \scriptsize]{l}{
}{\FL
  This table has all rows \NN
  set at script size \LL
}
```

Table 1: Doinside

This table has all rows
set at script size

11 Implementation

```
\RequirePackage{ifpdf,etoolbox,xcolor,xkeyval,array,tabularx,booktabs,rotating}
```

The transparency package works only in pdf mode, and if the tikz package is not loaded; otherwise define a dummy `\transparent` and issue a warning.

```
\ifpdf
  \@ifpackageloaded{tikz}{
    \PackageWarning{ctable}{
      Transparency disabled: incompatible with tikz package
    }
    \def\transparent#1{}
  }{
    \RequirePackage{transparent}
  }
\else
  \PackageWarningNoLine{ctable}{\MessageBreak
    Transparency disabled: pdfTeX is not running in PDF mode
  }
  \def\transparent#1{}
\fi
```

We need to know if the user has loaded tikz after ctable. If so, we have loaded the transparent package already, which then will disturb the tikz definitions, so we must quit with an error message. Some warnings depend on whether the caption package is loaded or not. Here a flag is set to remember that.

```
\newif\if@CTcaptionloaded
\AtBeginDocument{
  \makeatletter
  \@ifpackageloaded{tikz}{
    \@ifpackageloaded{transparent}{
      \PackageError{ctable}{You must load ctable after tikz}{}
    }
  }{}
  \@ifpackageloaded{caption}{\@CTcaptionloadedtrue}{\@CTcaptionloadedfalse}
  \makeatother
}
\def\NN{\tabularnewline}
\def\FL{\toprule}
\def\ML{\NN\midrule}
\def\LL{\NN\bottomrule}
\def\@dfltCTfgcolor#1 #2 #3={\definecolor{@dfltCTframefg}{rgb}{#1,#2,#3}}
\def\@dfltCTbgcolor#1 #2 #3={\definecolor{@dfltCTframebg}{rgb}{#1,#2,#3}}
\def\@CTfgcolor#1 #2 #3={%
  \definecolor{@CTframefg}{rgb}{#1,#2,#3}
  \def\@CTfgactual{@CTframefg}}
\def\@CTbgcolor#1 #2 #3={%
  \definecolor{@CTframebg}{rgb}{#1,#2,#3}
  \def\@CTbgactual{@CTframebg}}
\def\@CTtextsuperscript#1{%
  \ifx\@CTsuper\@CTtrue\@textsuperscript{#1}\else{\footnotesize#1}\fi
}
```

define a true and a false value

```
\def\@CTtrue{1}
\def\@CTfalse{0}
```

normally we do nothing special inside the float, but that can be changed with the doinside option

```
\def\@CTdoinside{\relax}
```

Need three booleans to remember: if we use tabularx, if we are running in the memoir class,

```
\newif\if@CTusex
\newif\if@CTinmemoir
\@ifclassloaded{memoir}{\@CTinmemoirtrue}{\@CTinmemoirfalse}
```

Need lots of dimens and their defaults

```
\newdimen\@CTframesep         \newdimen\@dfltCTframesep
\newdimen\@CTframerule        \newdimen\@dfltCTframerule
\newdimen\@CTwidth            \newdimen\@dfltCTwidth
\newdimen\@CTcaptionskip      \newdimen\@dfltCTcaptionskip
\newdimen\@CTmaxwidth         \newdimen\@dfltCTmaxwidth
\newdimen\@CTmincapwidth      \newdimen\@dfltCTmincapwidth
\newdimen\@CTfooterwidth      \newdimen\@dfltCTfooterwidth
\newdimen\@CTw % the final width
\newdimen\@CTfloatwidth
\newdimen\@CToldsep
\newdimen\@CToldrule
```

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

```
\newbox\CT@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 `\@CTbegin` we put `\begin{table}[tbp]` in it.

```

\define@key{suCT}{bgopacity}{\def\@dfltCTbgopacity{#1}}
\define@key{suCT}{botcap}[]{\let\@dfltCTbotcap\@CTtrue}
\define@key{suCT}{captionsinside}[]{\def\rot@LR{-1}
                                     \if@twoside\@rot@twosidetrue
                                     \else\@rot@twosidefalse\fi}
\define@key{suCT}{captionsleft}[]{\@rot@twosidefalse\def\rot@LR{-1}}
\define@key{suCT}{captionsright}[]{\@rot@twosidefalse\def\rot@LR{0}}
\define@key{suCT}{captionskip}{\@dfltCTcaptionskip=#1}
\define@key{suCT}{center}[]{\let\@dfltCTalign\centering}
\define@key{suCT}{continued}{\def\@dflttextcontinued{#1}}
\define@key{suCT}{doinside}{\def\@dfltCTdoinside{#1}}
\define@key{suCT}{figure}[]{\def\@dfltCTtaborfig{figure}}
\define@key{suCT}{framebg}{\@dfltCTbgcolor#1=}
\define@key{suCT}{framefg}{\@dfltCTfgcolor#1=}
\define@key{suCT}{framerule}{\@dfltCTframerule=#1}
\define@key{suCT}{framesep}{\@dfltCTframesep=#1}
\define@key{suCT}{left}[]{\let\@dfltCTalign\raggedright}
\define@key{suCT}{maxwidth}{\@dfltCTmaxwidth=#1}
\define@key{suCT}{mincapwidth}{\@dfltCTmincapwidth=#1}
\define@key{suCT}{footerwidth}[-1pt]{\@dfltCTfooterwidth=#1}
\define@key{suCT}{nonotespar}[]{\let\@dfltCTnotespar\@CTfalse}
\define@key{suCT}{nosideways}[]{\let\@dfltCTsideways\empty}
\define@key{suCT}{nostar}[]{\def\@dfltCTstarred{}}
\define@key{suCT}{nosuper}[]{\let\@dfltCTsuper\@CTfalse}
\define@key{suCT}{notespar}[]{\let\@dfltCTnotespar\@CTtrue}
\define@key{suCT}{pos}{\def\@dfltCTpos{#1}}
\define@key{suCT}{right}[]{\let\@dfltCTalign\raggedleft}
\define@key{suCT}{sideways}[]{\def\@dfltCTsideways{sideways}}
\define@key{suCT}{star}[]{\def\@dfltCTstarred{*}}
\define@key{suCT}{super}[]{\let\@dfltCTsuper\@CTtrue}
\define@key{suCT}{table}[]{\def\@dfltCTtaborfig{table}}
\define@key{suCT}{topcap}[]{\let\@dfltCTbotcap\@CTfalse}
\define@key{suCT}{width}{\@dfltCTwidth=#1}

\newcommand{\setupctable}[1]{\setkeys{suCT}{#1}}
\setupctable{
  bgopacity=1,
  captionskip=0pt,
  center,
  continued=(continued),
  doinside={},
  footerwidth=0pt,
  framebg=1 1 1,
  framefg=0 0 0,
  framerule=0pt,
  framesep=0pt,
  maxwidth=0pt,
  mincapwidth=0pt,
  nonotespar,
  nosideways,
  nostar,
  super,
  table,
  topcap,
  width=0pt,
}

\define@key{CT}{bgopacity}{\def\@CTbgopacity{#1}}
\define@key{CT}{botcap}[]{\let\@CTbotcap\@CTtrue}
\define@key{CT}{captionskip}{\@CTcaptionskip=#1}
\define@key{CT}{caption}{\def\@CTcaption{#1}}
\define@key{CT}{cap}{\def\@CTcap{#1}}
\define@key{CT}{center}[]{\let\@CTalign\centering}
\define@key{CT}{continued}[\@dflttextcontinued]{\def\@CTcontinued{#1}}
\define@key{CT}{doinside}{\def\@CTdoinside{#1}}
\define@key{CT}{figure}[]{\def\@CTtaborfig{figure}}
\define@key{CT}{framebg}{\@CTbgcolor#1=}
\define@key{CT}{framefg}{\@CTfgcolor#1=}
\define@key{CT}{framerule}{\@CTframerule=#1}
\define@key{CT}{framesep}{\@CTframesep=#1}
\define@key{CT}{label}{\def\@CTlabel{#1}}
\define@key{CT}{left}[]{\let\@CTalign\raggedright}
\define@key{CT}{maxwidth}{\@CTmaxwidth=#1}
\define@key{CT}{mincapwidth}{\@CTmincapwidth=#1}
\define@key{CT}{footerwidth}[-1pt]{\@CTfooterwidth=#1}
\define@key{CT}{nonotespar}[]{\let\@CTnotespar\@CTfalse}
\define@key{CT}{nosideways}[]{\let\@CTsideways\empty}
\define@key{CT}{nostar}[]{\def\@CTstarred{}}
\define@key{CT}{nosuper}[]{\let\@CTsuper\@CTfalse}
\define@key{CT}{notespar}[]{\let\@CTnotespar\@CTtrue}
\define@key{CT}{pos}{\def\@CTpos{#1}\def\@CTbegin{\@CTbeg[#1]}}
\define@key{CT}{right}[]{\let\@CTalign\raggedleft}
\define@key{CT}{sidecap}[]{\let\@CTbotcap\undefined}
\define@key{CT}{sideways}[]{\def\@CTsideways{sideways}}
\define@key{CT}{star}[]{\def\@CTstarred{*}}
\define@key{CT}{super}[]{\let\@CTsuper\@CTtrue}
\define@key{CT}{table}[]{\def\@CTtaborfig{table}}
\define@key{CT}{topcap}[]{\let\@CTbotcap\@CTfalse}
\define@key{CT}{width}{\@CTwidth=#1}
```

A caption will only be generated if the *caption* option was used, with a non-empty value. If so, it goes in the lot/lof, unless the *cap* option specified a different (probably shorter) value for it. A *cap* option with an empty value inhibits a tof/lof entry. The `\expandonce` trick below is from Marco Daniel. It expands the arguments of `\caption` so that the hyperref command `\nameref` works OK. See http://tex.stackexchange.com/questions/57396/ Note that, in captions, `tmark` may only be used without its optional argument.

```
\def\@CTCaption{
   \ifx\@CTcaption\empty\else
      \def\@CTcaptionarg{\ifx\@CTlabel\empty\else\label{\@CTlabel}\fi
         \@CTcaption\ \@CTcontinued\strut}
      \begingroup
        \ifx\@CTcap\empty
        \edef\x{\endgroup\noexpand\caption[]{\expandonce\@CTcaptionarg}}
      \else
      \edef\x{\endgroup\noexpand\caption[\expandonce\@CTcap]%
                                               {\expandonce\@CTcaptionarg}}
      \fi
    \x
  \fi
}
```

Need to redefine X columntype, but the array package would generate a warning. So first set the type to be redefined to `\undefined` to suppress the warning. Save the standard X type once in the new type Y

```
\newcolumntype{Y}{X}
\def\@CTXcolumntype#1{%
  \let\NC@find@X\undefined
  \newcolumntype{X}{#1}%
}
\long\def\@CTframe#1#2#3{%
  \@CToldsep\fboxsep\fboxsep\@CTframesep%
  \@CToldrule\fboxrule\fboxrule\@CTframerule%
  \transparent{\@CTbgopacity}%
  \fcolorbox{#1}{#2}{\fboxsep\@CToldsep\fboxrule\@CToldrule\transparent{1}#3}%
}
\newcommand{\tnote}[2][a]{%
  \ifx\@CTnotespar\@CTtrue%
    \@CTtextsuperscript{\normalfont\textit{#1}}\,#2
  \else%
    \hbox{\@CTtextsuperscript{\normalfont\textit{#1}}}&#2\NN
  \fi
}
\newcommand{\tmark}[1][a]{%
  \hbox{\textsuperscript{\normalfont\textit{#1}}}}
\newdimen\@CTcurftwidth
\newcommand{\ctable}[4][]{%
  \let\@CTtaborfig  \@dfltCTtaborfig
  \let\@CTalign     \@dfltCTalign
  \let\@CTsideways  \@dfltCTsideways
  \let\@CTcontinued \empty
  \let\@CTpos       \@dfltCTpos
  \let\@CTcaption   \empty
  \let\@CTcap       \undefined
  \let\@CTlabel     \empty
  \let\@CTbotcap    \@dfltCTbotcap
  \let\@CTstarred   \@dfltCTstarred
  \let\@CTsuper     \@dfltCTsuper
  \let\@CTnotespar  \@dfltCTnotespar
  \let\@CTdoinside  \@dfltCTdoinside
  \let\@CTbgopacity \@dfltCTbgopacity
  \@CTframerule     \@dfltCTframerule
  \@CTcaptionskip   \@dfltCTcaptionskip
  \@CTframesep      \@dfltCTframesep
  \@CTwidth         \@dfltCTwidth
  \@CTmaxwidth      \@dfltCTmaxwidth
  \@CTmincapwidth   \@dfltCTmincapwidth
  \@CTfooterwidth   \@dfltCTfooterwidth
  \def\@CTfgactual {@dfltCTframefg}%
  \def\@CTbgactual {@dfltCTframebg}%
  \def\@CTbeg      {\begin{\@CTsideways\@CTtaborfig\@CTstarred}}%
  \def\@CTbegin    {\@CTbeg}%
  \def\@CTend      {\end{\@CTsideways\@CTtaborfig\@CTstarred}}%
  \setkeys{CT}{#1}%
```

Make the short caption equal to the caption if it has not been defined

```
  \ifx\@CTcap\undefined\let\@CTcap\@CTcaption\fi
```

Issue a warning if the short caption is empty and the caption package is not loaded

```
  \ifx\@CTcap\empty
    \if@CTcaptionloaded\else
      \PackageWarningNoLine{ctable}{\MessageBreak
         An empty cap= option prevents lot/loc entry only\MessageBreak
         if the caption package is loaded!}
    \fi
  \fi
```

Currently, the sidecap option can only be used from within the memoir class; here we test if memoir is loaded:

```
\if@CTinmemoir\else
   \ifx\@CTbotcap\undefined
      \PackageError{ctable}{\MessageBreak
         You can, currently, use the sidecap option only with\MessageBreak
         memoir documents. Use topcap or botcap only}
   \fi
\fi
```

It makes no sense to use *width* together with *maxwidth* or *pos* together with *sideways*

```
\ifdim\@CTwidth=0pt\else
   \ifdim\@CTmaxwidth=0pt\else
      \PackageError{ctable}{\MessageBreak
         You may not use the width and maxwidth options together\MessageBreak
         Use either width or maxwidth}
   \fi
\fi
\ifx\@CTpos\empty
   \ifx\@CTsideways\empty\else
   \PackageError{ctable}{\MessageBreak
      You may not use the pos and sideways options together\MessageBreak
      Rotated tables and figures are always typeset on a separate page}
   \fi
\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 ??

```
\ifx\@CTcaption\empty
   \ifx\@CTlabel\empty\else
      \PackageError{ctable}{\MessageBreak
         You may not label a captionless table\MessageBreak
         Such a label can't be referenced}
   \fi
\fi
```

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

```
\sbox\CT@t{%
   \@CTXcolumntype{l}% temporarily make  type X = l
   \@CTframe{\@CTfgactual}{\@CTbgactual}{%
      \@CTdoinside
      \begin{tabular}{#2}
         #4%
      \end{tabular}%
   }%
}%
```

then look if we'll need the tabularx environment:

```
   \@CTusexfalse
   \ifdim\@CTmaxwidth=0pt
      \ifdim\@CTwidth=0pt
      \else
         \@CTusextrue
      \fi
   \else
      \ifdim\wd\CT@t>\@CTmaxwidth
         \@CTusextrue
      \fi
   \fi
%
% if so, replace tabular with tabularx:
%
```

```
   \if@CTusex
      \sbox\CT@t{%
         \@CTXcolumntype{Y}% restore X
         \@CTframe{\@CTfgactual}{\@CTbgactual}{%
            \@CTdoinside
            \begin{tabularx}{\ifdim\@CTwidth>0pt\@CTwidth\else\@CTmaxwidth\fi}{#2}
               #4%
            \end{tabularx}%
         }%
      }%
   \fi
```

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

```
   \@CTw=\wd\CT@t
```

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

```
   \@CTfloatwidth=\ifdim\@CTmincapwidth>\@CTw
      \@CTmincapwidth
   \else
      \@CTw
   \fi
```

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

```
   \@CTbegin
      \ifx\@CTcontinued\empty\else\addtocounter{\@CTtaborfig}{-1}\fi
      \@CTalign
      \begin{minipage}{\@CTfloatwidth}\parindent0pt
         \ifx\@CTbotcap\@CTfalse\@CTCaption\vskip\@CTcaptionskip\fi
         \ifx\@CTbotcap\undefined%
             \begin{sidecaption}[\@CTcap]{\@CTcaption}[\@CTlabel]
         \fi
         \centering{\usebox\CT@t}% insert the tabular
         \def\@CTfootnotes{#3}%
         \ifx#3\empty\else{% append footnotes, if any
```

Footnotes: if the `footerwidth` is 0pt (the default), typeset the footer as wide as the caption (which may be wider than the table because of the `mincapwidth` option); if it is -1pt (because `footerwidth` was set without an argument) make it as wide as the table; otherwise, give it the width set by the `footerwidth` option.

```
            \@CTcurftwidth=\ifdim\@CTfooterwidth=-1pt\@CTw\else
                                \ifdim\@CTfooterwidth=0pt\hsize\else
                                \@CTfooterwidth\fi\fi
            \footnotesize
            \ifx\@CTnotespar\@CTtrue%
               \\[.2ex]
               \begin{minipage}{\@CTcurftwidth}%
                  #3%
               \end{minipage}%
            \else%
               \\
               \begin{tabularx}{\@CTcurftwidth}{r@{\,}>{\raggedright}X}
                  #3%
               \end{tabularx}%
            \fi
         }
         \fi
         \ifx\@CTbotcap\undefined\end{sidecaption}\fi
         \ifx\@CTbotcap\@CTtrue\vskip\@CTcaptionskip\@CTCaption\fi
      \end{minipage}
   \@CTend
}
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.