

ctable — package for flexible key/value driven typesetting of floats

Wybo Dekker*

v1.33 2025/10/22

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 Purpose	1
2 Usage	2
3 Options	2
4 The width and maxwidth options	4
5 Tables wider than the text width	4
6 Setting option defaults: <code>setupctable</code>	4
7 Other commands	5
8 Examples	6
9 Option examples	7
9.1 <code>center, left, right</code>	7
9.2 <code>super, nosuper</code>	8
9.3 <code>notespar, nonotespar</code>	8
9.4 <code>continued</code>	8
9.5 <code>mincapwidth</code>	9
9.6 <code>maxwidth</code>	9
9.7 <code>framerule</code>	10
9.8 <code>captionskip</code>	11
9.9 <code>figure, botcap</code>	11
9.10 <code>doinside</code>	12
10 Implementation	12
10.1 Initialization	12
11 History	19
12 Index	21

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.

*Email: wybodekker@me.com

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`, 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.

2 Usage

`\ctable[key=value,...]{coldefs}{foottable}{rows}`

typesets the *rows* in a table float with column specifiers *coldefs* like in the `tabularx` environment. *foottable* consists of zero or more instances of the `\tnote` command described in section 7. The table properties are specified with *key=value* pairs described in section 3.

`\setupctable[key=value,...]`

sets `\ctable` defaults, either in the preamble or in the body.

3 Options

Options are given as *key=value* pairs, separated by comma's. Extra comma's, including any 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:

`bgopacity (env)...`

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 (env)` put the caption at the bottom of the float instead of on top of it. See also: `topcap`, `sidecap`.

`caption (env)...`

table caption; the braces are needed only if your caption contains a comma or an equals sign.

`cap (env)...`

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...`

(env.) 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`

¹I 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

class, memoir's caption commands are suspended and caption's commands must be used.

- `captionsleft` (env.) 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` (env.) 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` (env.) 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` (env.) center the table in the available text width; this is the default. See also: `left`, `right`.
- `continued` (env) ...
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` (env) ...
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` (env.) produce a figure float instead of a table float. See also: `table`.
- `footerwidth` ...
(env.) 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 the table.
- `framebg` (env) $\rightarrow r\ g\ b$
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` (env) $\rightarrow r\ g\ b$
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` (env) ...
draw a frame around the table with the given rule thickness. The default is 0pt, so that no frame will be seen.
- `framesep` (env) ...
set the distance between the frame and the table to the given dimension. The default is 0pt.
- `label` (env) ...
labels the float with `\label`.
- `left` (env.) left align the table in the available text width. See also: `center`, `right`.
- `maxwidth` (env) ...
like the *width* option, but any X column specifiers will be replaced with 1 if the resulting table width would thus stay within the specified maximum width. This is especially useful where the L^AT_EX source is generated by a script.
- `mincapwidth` ...
(env.) 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` (env.) typeset footnotes in a table; this is the default. See also: `notespar`.
- `nosideways` (env.) undo the `sideways` option. See also: `sideways`.
- `nostar` (env.) use the un-starred versions of the `table` and `figure` environments; this is the default
- `nosuper` (env.) in the footnote table, typeset footnote markers on the line, instead of superscripted.

`notespar (env.)` typeset footnotes in a paragraph instead of in a table.

`pos (env.)` . . .
float position, default: `tbp`.

`right (env.)` right align the table in the available text width.

`sidecap (env.)` put the caption at the side of the float. Currently, this works only if you have loaded the `memoir` 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 (env.)` 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 (env.)` 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 (env.)` in the footnote table, typeset footnote markers as superscripts; this is the default. See also: `nosuper`.

`table (env.)` produce a table float (this is the default). See also: `figure`.

`topcap (env.)` put the caption top of the float; this is the default. See also: `botcap`, `sidecap`.

`width (env.)` . . .
`tabularx` will be used to typeset the table at the specified width — one or more `X` column specifiers must be provided.

4 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.

5 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`.

6 Setting option defaults: `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` `\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
}

```

7 Other commands

`\tnote[label]{footnote text}`

places ^{*label*} footnote text under the table. This command can only be used in `\ctable`'s third argument, that is: 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[label]` 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, that is: in tabulars, longtables, et cetera.

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.

8 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
             $\text{Mu(H)} + \text{X}_2$  and  $\text{Mu(H)} + \text{HX}$ ~\tmark,
  label    = nowidth,
  pos      = h
]{rlcc}{
  \tnote{for the abstraction reaction,
     $\text{Mu} + \text{HX} \rightarrow \text{MuH} + \text{X}$ .}
  \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.}
}{
  & &  $\text{H(Mu)} + \text{F}_2$  &  $\text{H(Mu)} + \text{Cl}_2$  & \FL
  &  $\beta(\text{H})$  &  $80.9^\circ$ \tmark[b] &  $83.2^\circ$  & \ML
  &  $\beta(\text{Mu})$  &  $86.7^\circ$  &  $87.7^\circ$  & \LL
}
```

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

	H(Mu) + F ₂	H(Mu) + 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.

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}
}{
  \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}
  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 9.9 on page 11.

9 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.

9.1 center, left, right

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

<pre>\ctable[caption = Centered,]{c}{c}{\FL Table's first row\L</pre>	<div>Table 1: Centered</div> <div>Table's first row</div>
<pre>\ctable[caption = Left, left]{c}{c}{\FL Table's first row\L</pre>	<div>Table 1: Left</div> <div>Table's first row</div>
<pre>\ctable[caption = Right, right]{c}{c}{\FL Table's first row\L</pre>	<div>Table 1: Right</div> <div>Table's first row</div>

9.2 `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

9.3 `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 a 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 a 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 perceive 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.

9.4 `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

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

Table 1: Caption (*contd*)

Table's first row

9.5 mincapwidth

`cetable` 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:

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

Table 1:

a
lengthy
caption

row1

```
\cetable[
  mincapwidth = 40mm,
  caption = a lengthy caption
]{c}{\tnote{this is a footnote}}
{\FL row1\mark\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:

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

Table 1: a lengthy caption

row1 ^a

^a footnote

9.6 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 1 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 1 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
---	------

9.7 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 background 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

9.8 captionskip

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

<pre>\ctable[caption = Caption,]{c}{\FL Table's first row\LL}</pre>	<table border="1"> <tr><td>Table 1: Caption</td></tr> <tr><td>Table's first row</td></tr> </table>	Table 1: Caption	Table's first row
Table 1: Caption			
Table's first row			

<pre>\ctable[caption = Caption, captionskip = 1ex,]{c}{\FL Table's first row\LL}</pre>	<table border="1"> <tr><td>Table 1: Caption</td></tr> <tr><td>Table's first row</td></tr> </table>	Table 1: Caption	Table's first row
Table 1: Caption			
Table's first row			

This works for bottom caption, too:

<pre>\ctable[caption = Caption, botcap]{c}{\FL Table's first row\LL}</pre>	<table border="1"> <tr><td>Table's first row</td></tr> <tr><td>Table 1: Caption</td></tr> </table>	Table's first row	Table 1: Caption
Table's first row			
Table 1: Caption			

<pre>\ctable[caption = Caption, captionskip = -2ex, botcap]{c}{\FL Table's first row\LL}</pre>	<table border="1"> <tr><td>Table's first row</td></tr> <tr><td>Table 1: Caption</td></tr> </table>	Table's first row	Table 1: Caption
Table's first row			
Table 1: Caption			

9.9 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:

<pre>\ctable[caption = a table]{c}{ }{\FL Table's first row\LL}</pre>	<table border="1"> <tr><td>Table 1: a table</td></tr> <tr><td>Table's first row</td></tr> </table>	Table 1: a table	Table's first row
Table 1: a table			
Table's first row			

```
\newcommand{\F}[1]{
  \includegraphics[width=\hsize]{#1}
}
\newcolumnntype{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
}
```

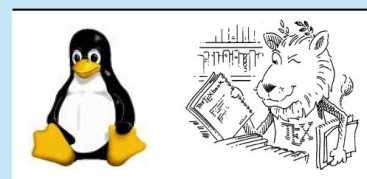


Figure 1: a figure

9.10 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]{1}{
}{\FL
  This table has all rows \NN
  set at script size \LL
}
```

Table 1: Doinside

This table has all rows set at script size

Put your descriptive text here

10 Implementation

10.1 Initialization

```
1 (*package)
2 \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.

```
3 \ifpdf
4   \ifpackageloaded{tikz}{
5     \PackageWarningNoLine{ctable}{
6       Transparency disabled: incompatible with tikz package
7     }
8     \def\transparent#1{}
9   }{
10    \RequirePackage{transparent}
11  }
12 \else
13   \PackageWarningNoLine{ctable}{\MessageBreak
14     Transparency disabled: pdfTeX is not running in PDF mode
15   }
16   \def\transparent#1{}
17 \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.

```
18 \newif\if@CTcaptionloaded
19 \AtBeginDocument{%
20   \ifpackageloaded{tikz}{%
21     \ifpackageloaded{transparent}{%
22       \PackageError{ctable}{You must load ctable after tikz}{}%
23     }{}%
24   }{}%
25   \ifpackageloaded{caption}{\@CTcaptionloadedtrue}{\@CTcaptionloadedfalse}%
26 }
27 \def\NN{\tabularnewline}
28 \def\FL{\toprule}
29 \def\ML{\NN\midrule}
30 \def\LL{\NN\bottomrule}
31 \def\@dfltCTfgcolor#1 #2 #3={\definecolor{@dfltCTframefg}{rgb}{#1,#2,#3}}
32 \def\@dfltCTbgcolor#1 #2 #3={\definecolor{@dfltCTframebg}{rgb}{#1,#2,#3}}
33 \def\@CTfgcolor#1 #2 #3={%
34   \definecolor{CTframefg}{rgb}{#1,#2,#3}
35   \def\@CTfgactual{CTframefg}}
36 \def\@CTbgcolor#1 #2 #3={%
37   \definecolor{CTframebg}{rgb}{#1,#2,#3}
38   \def\@CTbgactual{CTframebg}}
39 \def\@CTtextsuperscript#1{%
40   \ifx\@CTsuper\@CTtrue\@textsuperscript{#1}\else\footnotesize#1\fi
```

```

41 }
define a true and a false value
42 \def\@CTtrue{1}
43 \def\@CTfalse{0}

normally we do nothing special inside the float, but that can be changed with the doinside option
44 \def\@CTdoincide{\relax}

Need three booleans to remember: if we use tabularx, if we are running in the memoir class,
45 \newif\if@CTusex
46 \newif\if@CTinmemoir
47 \@ifclassloaded{memoir}{\@CTinmemoirtrue}{\@CTinmemoirfalse}

Need lots of dimens and their defaults
48 \newdimen\@CTframesep          \newdimen\@dfloatCTframesep
49 \newdimen\@CTframerule         \newdimen\@dfloatCTframerule
50 \newdimen\@CTwidth             \newdimen\@dfloatCTwidth
51 \newdimen\@CTcaptionkip        \newdimen\@dfloatCTcaptionkip
52 \newdimen\@CTmaxwidth          \newdimen\@dfloatCTmaxwidth
53 \newdimen\@CTmincapwidth       \newdimen\@dfloatCTmincapwidth
54 \newdimen\@CTfooterwidth       \newdimen\@dfloatCTfooterwidth
55 \newdimen\@CTw % the final width
56 \newdimen\@CTfloatwidth
57 \newdimen\@CToldsep
58 \newdimen\@CToldrule

Allocate box registers so that we can determine the widths of the tables
59 \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.
60 \define@key{suCT}{bgopacity}{\def\@dfloatCTbgopacity{#1}}
61 \define@key{suCT}{botcap}[]{\let\@dfloatCTbotcap\@CTtrue}
62 \define@key{suCT}{captionsinside}[]{\def\rot@LR{-1}}
63                                     \if@twoside\@rot@twosidetrue
64                                     \else\@rot@twosidefalse\fi
65 \define@key{suCT}{captionsleft}[]{\@rot@twosidefalse\def\rot@LR{-1}}
66 \define@key{suCT}{captionsright}[]{\@rot@twosidefalse\def\rot@LR{0}}
67 \define@key{suCT}{captionkip}{\@dfloatCTcaptionkip=#1}
68 \define@key{suCT}{center}[]{\let\@dfloatCTalign\centering}
69 \define@key{suCT}{continued}{\def\@dfloattextcontinued{#1}}
70 \define@key{suCT}{doincide}{\def\@dfloatCTdoincide{#1}}
71 \define@key{suCT}{figure}[]{\def\@dfloatCTaborfig{figure}}
72 \define@key{suCT}{framebg}{\@dfloatCTbgcolor#1=}
73 \define@key{suCT}{framefg}{\@dfloatCTfgcolor#1=}
74 \define@key{suCT}{framerule}{\@dfloatCTframerule=#1}
75 \define@key{suCT}{framesep}{\@dfloatCTframesep=#1}
76 \define@key{suCT}{left}[]{\let\@dfloatCTalign\raggedright}
77 \define@key{suCT}{maxwidth}{\@dfloatCTmaxwidth=#1}
78 \define@key{suCT}{mincapwidth}{\@dfloatCTmincapwidth=#1}
79 \define@key{suCT}{footerwidth}{-1pt}{\@dfloatCTfooterwidth=#1}
80 \define@key{suCT}{nonotespar}[]{\let\@dfloatCTnotespar\@CTfalse}
81 \define@key{suCT}{nosideways}[]{\let\@dfloatCTsideways\empty}
82 \define@key{suCT}{nostar}[]{\def\@dfloatCTstarred{}}
83 \define@key{suCT}{nosuper}[]{\let\@dfloatCTsuper\@CTfalse}
84 \define@key{suCT}{notespar}[]{\let\@dfloatCTnotespar\@CTtrue}
85 \define@key{suCT}{pos}{\def\@dfloatCTpos{#1}}
86 \define@key{suCT}{right}[]{\let\@dfloatCTalign\raggedleft}
87 \define@key{suCT}{sideways}[]{\def\@dfloatCTsideways{sideways}}
88 \define@key{suCT}{star}[]{\def\@dfloatCTstarred{*}}
89 \define@key{suCT}{super}[]{\let\@dfloatCTsuper\@CTtrue}
90 \define@key{suCT}{table}[]{\def\@dfloatCTaborfig{table}}
91 \define@key{suCT}{topcap}[]{\let\@dfloatCTbotcap\@CTfalse}
92 \define@key{suCT}{width}{\@dfloatCTwidth=#1}
93 \newcommand\setupctable[1]{\setkeys{suCT}{#1}}
94 \setupctable{
95   bgopacity=1,
96   captionkip=0pt,

```

```

97   center,
98   continued=(continued),
99   doinside={},
100  footerwidth=0pt,
101  framebg=1 1 1,
102  framefg=0 0 0,
103  framerule=0pt,
104  framesep=0pt,
105  maxwidth=0pt,
106  mincapwidth=0pt,
107  nonotespar,
108  nosideways,
109  nostar,
110  super,
111  table,
112  topcap,
113  width=0pt,
114 }
115 \define@key{CT}{bgopacity}{\def\@CTbgopacity{#1}}
116 \define@key{CT}{botcap}[]{\let\@CTbotcap\@CTtrue}
117 \define@key{CT}{captionskip}{\@CTcaptionskip=#1}
118 \define@key{CT}{caption}{\def\@CTcaption{#1}}
119 \define@key{CT}{cap}{\def\@CTcap{#1}}
120 \define@key{CT}{center}[]{\let\@CTalign\centering}
121 \define@key{CT}{continued}[\@dflltttextcontinued]{\def\@CTcontinued{#1}}
122 \define@key{CT}{doininside}{\def\@CTdoininside{#1}}
123 \define@key{CT}{figure}[]{\def\@CTtaborfig{figure}}
124 \define@key{CT}{framebg}{\@CTbgcolor=#1}
125 \define@key{CT}{framefg}{\@CTfgcolor=#1}
126 \define@key{CT}{framerule}{\@CTframerule=#1}
127 \define@key{CT}{framesep}{\@CTframesep=#1}
128 \define@key{CT}{label}{\def\@CTlabel{#1}}
129 \define@key{CT}{left}[]{\let\@CTalign\raggedright}
130 \define@key{CT}{maxwidth}{\@CTmaxwidth=#1}
131 \define@key{CT}{mincapwidth}{\@CTmincapwidth=#1}
132 \define@key{CT}{footerwidth}[-1pt]{\@CTfooterwidth=#1}
133 \define@key{CT}{nonotespar}[]{\let\@CTnotespar\@CTfalse}
134 \define@key{CT}{nosideways}[]{\let\@CTsideways\empty}
135 \define@key{CT}{nostar}[]{\def\@CTstarred{}}
136 \define@key{CT}{nosuper}[]{\let\@CTsuper\@CTfalse}
137 \define@key{CT}{notespar}[]{\let\@CTnotespar\@CTtrue}
138 \define@key{CT}{pos}{\def\@CTpos{#1}\def\@CTbegin{\@CTbeg{#1}}}
139 \define@key{CT}{right}[]{\let\@CTalign\raggedleft}
140 \define@key{CT}{sidecap}[]{\let\@CTbotcap\undefined}
141 \define@key{CT}{sideways}[]{\def\@CTsideways{sideways}}
142 \define@key{CT}{star}[]{\def\@CTstarred{*}}
143 \define@key{CT}{super}[]{\let\@CTsuper\@CTtrue}
144 \define@key{CT}{table}[]{\def\@CTtaborfig{table}}
145 \define@key{CT}{topcap}[]{\let\@CTbotcap\@CTfalse}
146 \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 `\nameref` command works OK. See <http://tex.stackexchange.com/questions/57396/> Note that, in captions, `tmark` may only be used without its optional argument.

```

147 \def\@CTCaption{
148   \ifx\@CTcaption\empty\else
149     \def\@CTcaptionarg{\ifx\@CTlabel\empty\else\label{\@CTlabel}\fi
150       \@CTcaption\ \@CTcontinued\strut}
151     \begingroup
152       \ifx\@CTcap\empty
153         \edef\x{\endgroup\noexpand\caption[]{\expandonce\@CTcaptionarg}}
154       \else
155         \edef\x{\endgroup\noexpand\caption[\expandonce\@CTcap]%

```

```

156                                     {\expandonce\@CTcaptionarg}}
157     \fi
158     \x
159     \fi
160 }

```

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

```

161 \newcolumntype{Y}{X}
162 \def\@CTXcolumntype#1{%
163     \let\NC@find@X\undefined
164     \newcolumntype{X}{#1}%
165 }
166 \long\def\@CTframe#1#2#3{%
167     \@CToldsep\fbboxsep\fbboxsep\@CTframesep%
168     \@CToldrule\fbboxrule\fbboxrule\@CTframerule%
169     \transparent{\@CTbgopacity}%
170     \fcolorbox{#1}{#2}{\fbboxsep\@CToldsep\fbboxrule\@CToldrule\transparent{1}#3}%
171 }
172 \newcommand{\tnote}[2][a]{%
173     \ifx\@CTnotespar\@CTtrue%
174         \@CTtextsuperscript{\normalfont\textit{#1}}\,,#2
175     \else%
176         \hbox{\@CTtextsuperscript{\normalfont\textit{#1}}}&#2\NN
177     \fi
178 }
179 \newcommand{\tmark}[1][a]{%
180     \hbox{\textsuperscript{\normalfont\textit{#1}}}}
181 \newdimen\@CTcurftwidth
182 \newcommand{\ctable}[4][[]]{%
183     \let\@CTtaborfig \@dfltCTtaborfig
184     \let\@CTalign \@dfltCTalign
185     \let\@CTsideways \@dfltCTsideways
186     \let\@CTcontinued \empty
187     \let\@CTpos \@dfltCTpos
188     \let\@CTcaption \empty
189     \let\@CTcap \undefined
190     \let\@CTlabel \empty
191     \let\@CTbotcap \@dfltCTbotcap
192     \let\@CTstarred \@dfltCTstarred
193     \let\@CTsuper \@dfltCTsuper
194     \let\@CTnotespar \@dfltCTnotespar
195     \let\@CTdoinside \@dfltCTdoinside
196     \let\@CTbgopacity \@dfltCTbgopacity
197     \@CTframerule \@dfltCTframerule
198     \@CTcaptionskip \@dfltCTcaptionskip
199     \@CTframesep \@dfltCTframesep
200     \@CTwidth \@dfltCTwidth
201     \@CTmaxwidth \@dfltCTmaxwidth
202     \@CTmincapwidth \@dfltCTmincapwidth
203     \@CTfooterwidth \@dfltCTfooterwidth
204     \def\@CTfgactual {\@dfltCTframefg}%
205     \def\@CTbgactual {\@dfltCTframebg}%
206     \def\@CTbeg {\begin{\@CTsideways\@CTtaborfig\@CTstarred}}%
207     \def\@CTbegin {\@CTbeg}%
208     \def\@CTend {\end{\@CTsideways\@CTtaborfig\@CTstarred}}%
209     \setkeys{CT}{#1}%

```

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

```

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

```

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

```

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

```

```

216     }
217     \fi
218     \fi

```

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

```

219     \if@CTinmemoir\else
220         \ifx\@CTbotcap\undefined
221             \PackageError{ctable}%
222                 {sidecap option not available here}%
223                 {\MessageBreak You can, currently, use the sidecap option only with
224                 \MessageBreak memoir documents. Use topcap or botcap only.
225             }
226     \fi
227     \fi

```

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

```

228     \ifdim\@CTwidth=0pt\else
229         \ifdim\@CTmaxwidth=0pt\else
230             \PackageError{ctable}
231                 {width and maxwidth options are mutually exclusive}
232                 {\MessageBreak You may not use the width and maxwidth options together.
233                 \MessageBreak Use either width or maxwidth.
234             }
235         \fi
236     \fi
237     \ifx\@CTpos\undefined\else
238         \ifx\@CTsideways\empty\else
239             \PackageError{ctable}
240                 {pos and sideways options are mutually exclusive}
241                 {\MessageBreak You may not use the pos and sideways options together.
242                 \MessageBreak Rotated tables and figures appear on a separate page.
243             }
244         \fi
245     \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 ??

```

246     \ifx\@CTcaption\empty
247         \ifx\@CTlabel\empty\else
248             \PackageError{ctable}
249                 {label option used in captionless table}
250                 {\MessageBreak You may not label a captionless table.
251                 \MessageBreak Such a label can't be referenced.
252             }
253         \fi
254     \fi

```

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

```

255     \sbox\CT@t{%
256         \@CTXcolumnntype{1}% temporarily make type X = 1
257         \@CTframe{\@CTfgactual}{\@CTbgactual}{%
258             \@CTdoinside
259             \begin{tabular}{#2}
260                 #4%
261             \end{tabular}%
262         }%
263     }%

```

then look if we'll need the `tabularx` environment:

```

264     \@CTusexfalse
265     \ifdim\@CTmaxwidth=0pt
266         \ifdim\@CTwidth=0pt
267             \else
268                 \@CTusextrue
269             \fi
270     \else
271         \ifdim\wd\CT@t>\@CTmaxwidth

```



```

272     \@CTusextrue
273     \fi
274     \fi
275 %
276 % if so, replace tabular with tabularx:
277 %
278     \if@CTusex
279         \sbox\CT@t{%
280             \@CTXcolumnntype{Y}% restore X
281             \@CTframe{\@CTfgactual}{\@CTbgactual}{%
282                 \@CTdoinside
283                 \begin{tabularx}{\ifdim\@CTwidth>0pt\@CTwidth\else\@CTmaxwidth\fi}{#2}
284                     #4%
285                 \end{tabularx}%
286             }%
287         }%
288     \fi

```

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

```

289     \@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:

```

290     \@CTfloatwidth=\ifdim\@CTmincapwidth>\@CTw
291         \@CTmincapwidth
292     \else
293         \@CTw
294     \fi

```

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

```

295     \@CTbegin
296         \ifx\@CTcontinued\empty\else\addtocounter{\@CTtaborfig}{-1}\fi
297         \@CTalign
298         \begin{minipage}{\@CTfloatwidth}\parindent0pt
299             \ifx\@CTbotcap\@CTfalse\@CTCaption\vskip\@CTcaptionskip\fi
300             \ifx\@CTbotcap\undefined%
301                 \begin{sidecaption}[\@CTcap]{\@CTcaption}[\@CTlabel]
302                 \fi
303                 \centering{\usebox\CT@t}% insert the tabular
304             \def\@CTfootnotes{#3}%
305             \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.

```

306                 \@CTcurftwidth=\ifdim\@CTfooterwidth=-1pt\@CTw\else
307                     \ifdim\@CTfooterwidth=0pt\hsize\else
308                         \@CTfooterwidth\fi\fi
309                 \footnotesize
310                 \ifx\@CTnotespar\@CTtrue%
311                     \[.2ex]
312                     \begin{minipage}{\@CTcurftwidth}%
313                         #3%
314                     \end{minipage}%
315                 \else%
316                     \[
317                     \begin{tabularx}{\@CTcurftwidth}{r@{\,}>{\raggedright}X}
318                         #3%
319                     \end{tabularx}%
320                 \fi
321             }
322         \fi
323         \ifx\@CTbotcap\undefined\end{sidecaption}\fi
324         \ifx\@CTbotcap\@CTtrue\vskip\@CTcaptionskip\@CTCaption\fi
325     \end{minipage}
326     \@CTend
327 }

```

```
328 \endinput
329 \end{package}
```

11 History

- v1.00 2000-06-01
 - first release.
- v1.01 2001-03-17
 - making use of booktabs package
- v1.02 2002-06-24
 - using keyval to reduce args to 4
- v1.03 2002-07-16
 - many syntactic corrections, thanks to Johannes Braams
- v1.04 2003-08-11
 - caption, if empty, will not be typeset
 - rotate option added
 - star option added to use table* and figure*
 - environments
- v1.05 2003-10-03
 - maxwidth option added
- v1.06 2004-03-20
 - left, right and center options added
 - framesep,rule,fg,bg options added
 - error in width-setting corrected
- v1.06a 2004-04-01
 - made setting fboxsep and fboxrule only temporary
 - removed superfluous space after tabulars
- v1.06b 2004-06-19
 - added several % at eol to remove superfluous whitespace occurring sometimes
- v1.07 2005-08-09
 - added option sideways, option rotate now obsolete
 - added option captionskip
- v1.08 2006-04-10
 - standardized file setup following dtxut.pdf
 - mincapwidth option added
 - moved newdimen definition outside ctable macro
- v1.09 2007-03-04
 - added option nosuper
 - corrected incorrect positioning when table is wider than mincapwidth
- v1.10 2007-08-17
 - 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
 - captionskip option redefined: Opt value now corresponds to LaTeX default
- v1.11 2007-09-07
 - added some percent signs at EOL to prevent whitespace
 - removed xspace usage - caused overfull badness
- v1.12 2008-04-12
 - option notespar added
- v1.13 2008-05-01
 - cap option with empty argument will not be inserted in lot/lof
 - added option continued, for continuation tables
- v1.14 2009-09-15
 - nosuper propagation to later tables prohibited
 - added option doinside
 - use of (obsolete) carom.sty for docs discontinued
 - empty labels not created
 - newcolumnntype warnings removed
 - caption package not needed anymore
- v1.15 2009-09-17
 - removed whitespace before tables
 - corrected marginpars in the documentation

- v1.16 2010-06-26
 - option cap= did not suppress lot/lof entry
 - notespar option now generates fully justified notes
- v1.17 2010-10-30
 - doinside option propagated in subsequent ctable calls
- v1.18 - 2011-04-15
 - added setupctable for option defaults
 - added complement for several options (topcap, nosideways, et cetera)
- v1.19 2011-05-01
 - sideways option did not work anymore; corrected
- v1.20 2011-08-24
 - added options captionsleft, captionsright, captionsinside
- v1.21 2011-09-05
 - better documentation for sideways, captionsleft/right/inside options
- v1.22 2012-05-25
 - allow empty lines in last (tabular) argument
 - corrected error from hyperref's nameref calls (thanks Marco Daniel!)
- v1.23 2012-05-28
 - footerwidth option added
- v1.24 - 2013-04-28
 - require xcolor instead of color
 - added option bgopacity
 - added option sidecap (for memoir only)
- v1.25 2013-05-24
 - url's to CTAN corrected
- v1.26 2013-06-15
 - footerwidth option was inactive when notespar option was active
- v1.27 - 2013-12-19
 - label option did not work with side caption
 - disable transparency with warning if tikz package loaded
- v1.28 2014-02-20
 - added percent characters at EOLs causing whitespace
- v1.29 2014-04-20
 - reorganized inst script
- v1.30 2015-08-29
 - tikz' transparency got broken when tikz loaded after ctable
 - removed overfull hboxes from documentation
- v1.31 2015-10-03
 - documentation: comments about tmark in captions
 - documentation: more info about caption skips
 - use expandonce from etoolbox
 - use AtBeginDocument, not AtEndPreamble from etoolbox (problems with tikz)
- v1.32 2025-07-10
 - help argument missing in PackageError calls
 - simultaneous use of pos and sideways options incorrectly tested
 - now contained in 1 self-extracting file
- v1.33 - missing args inside AtBeginDocument corrected

12 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.

Symbols			D
\,	174, 317	\@CTtaborfig	123, 144, 183, 206, 208, 296
\@CTCaption . . .	147, 299, 324	\@CTtextsuperscript	39, 174, 176
\@CTXcolumn type	162, 256, 280	\@CTtrue	40, 42, 61, 84, 89, 116, 137, 143, 173, 310, 324
\@CTalign	120, 129, 139, 184, 297	\@CTusefalse	264
\@CTbeg	138, 206, 207	\@CTusetrue	268, 272
\@CTbegin	138, 207, 295	\@CTw	55, 289, 290, 293, 306
\@CTbgactual . . .	38, 205, 257, 281	\@CTwidth	50, 146, 200, 228, 266, 283
\@CTbgcolor	36, 124	\@dfltCTalign	68, 76, 86, 184
\@CTbgopacity . .	115, 169, 196	\@dfltCTbgcolor	32, 72
\@CTbotcap	116, 140, 145, 191, 220, 299, 300, 323, 324	\@dfltCTbgopacity . .	60, 196
\@CTcap	119, 152, 155, 189, 210, 211, 301	\@dfltCTbotcap . . .	61, 91, 191
\@CTcaption	118, 148, 150, 188, 210, 246, 301	\@dfltCTcaption skip	51, 67, 198
\@CTcaptionarg . .	149, 153, 156	\@dfltCTdoinside . . .	70, 195
\@CTcaptionloadedfalse .	25	\@dfltCTfgcolor	31, 73
\@CTcaptionloadedtrue .	25	\@dfltCTfooterwidth	54, 79, 203
\@CTcaption skip	51, 117, 198, 299, 324	\@dfltCTframerule . . .	49, 74, 197
\@CTcontinued	121, 150, 186, 296	\@dfltCTframesep . . .	48, 75, 199
\@CTcurftwidth	181, 306, 312, 317	\@dfltCTmaxwidth . . .	52, 77, 201
\@CTdoinside	44, 122, 195, 258, 282	\@dfltCTmincapwidth	53, 78, 202
\@CTend	208, 326	\@dfltCTnotespar	80, 84, 194
\@CTfalse	43, 80, 83, 91, 133, 136, 145, 299	\@dfltCTpos	85, 187
\@CTfgactual	35, 204, 257, 281	\@dfltCTsideways . . .	81, 87, 185
\@CTfgcolor	33, 125	\@dfltCTstarred	82, 88, 192
\@CTfloatwidth . . .	56, 290, 298	\@dfltCTsuper	83, 89, 193
\@CTfooterwidth	54, 132, 203, 306, 307, 308	\@dfltCTtaborfig . . .	71, 90, 183
\@CTfootnotes	304	\@dfltCTwidth	50, 92, 200
\@CTframe	166, 257, 281	\@dflttextcontinued . .	69, 121
\@CTframerule . . .	49, 126, 168, 197	\\	311, 316
\@CTframesep . . .	48, 127, 167, 199	_	150
\@CTinmemoirfalse	47		
\@CTinmemoirtrue	47	A	
\@CTlabel	128, 149, 190, 247, 301	\addlinespace	5
\@CTmaxwidth	52, 130, 201, 229, 265, 271, 283		
\@CTmincapwidth	53, 131, 202, 290, 291	B	
\@CTnotespar	133, 137, 173, 194, 310	bgopacity (env.)	2
\@CToldrule	58, 168, 170	botcap (env.)	2
\@CToldsep	57, 167, 170	\bottomrule	5, 30
\@CTpos	138, 187, 237		
\@CTsideways	134, 141, 185, 206, 208, 238	C	
\@CTstarred	135, 142, 192, 206, 208	cap (env.)	2
\@CTsuper	40, 136, 143, 193	\caption	153, 155
		caption (env.)	2
		captionsinside (env.)	3
		caption skip (env.)	2
		captionsleft (env.)	3
		captionsright (env.)	3
		center (env.)	3
		\centering	68, 120, 303
		\cmidrule	5
		continued (env.)	3
		\CT@t	59, 255, 271, 279, 289, 303
		\ctable	2, 182
		E	
		environments:	
		bgopacity	2
		botcap	2
		cap	2
		caption	2
		captionsinside	3
		caption skip	2
		captionsleft	3
		captionsright	3
		center	3
		continued	3
		doinside	3
		figure	3
		footerwidth	3
		framebg	3
		framefg	3
		framerule	3
		framesep	3
		label	3
		left	3
		maxwidth	3
		mincapwidth	3
		notespar	3
		nosideways	3
		nostar	3
		nosuper	3
		notespar	4
		pos	4
		right	4
		sidecap	4
		sideways	4
		star	4
		super	4
		table	4
		topcap	4
		width	4
		F	
		figure (env.)	3
		\FL	5, 28
		footerwidth (env.)	3
		\footnotesize	40, 309
		framebg (env.)	3
		framefg (env.)	3
		framerule (env.)	3
		framesep (env.)	3
		I	
		\if@CTcaptionloaded	18, 212
		\if@CTinmemoir	46, 219
		\if@CTusex	45, 278
		L	
		label (env.)	3
		left (env.)	3
		\\	5, 30

M		P		T	
maxwidth (env.)	3	pos (env.)	4	table (env.)	4
\midrule	5, 29			\tabularnewline	27
mincapwidth (env.)	3	R		\tmark	5, 179
\ML	5, 29	right (env.)	4	\tnote	5, 172
\morecmidrules	5	S		topcap (env.)	4
N		\setkeys	93, 209	\toprule	5, 28
\NN	5, 27, 29, 30, 176	\setupctable	2, 4, 93, 94	\transparent	8, 16, 169, 170
nonotespar (env.)	3	sidecap (env.)	4	W	
nosideways (env.)	3	sideways (env.)	4	width (env.)	4
nostar (env.)	3	\specialrule	5	X	
nosuper (env.)	3	star (env.)	4	\x	153, 155, 158
notespar (env.)	4	super (env.)	4		