The `ctable` package*
for use with LaTeX2e

Wybo Dekker
wybo@servalys.nl

2004/06/19

Contents

List of Tables

List of Figures

1 Purpose

The `ctable` package lets you easily typeset captioned table and figure floats with optional footnotes. Both caption and footnotes will be forced within the width of the table.
If the width of the table is specified, then tabularx will be used to typeset it, and the `X` column specifier can be used. Otherwise tabular will be used.
This package defines the commands `\ctable`, `\tnote` and `\tmark`, as well as four `\tabularnewline` generating commands. The latter generate reasonable amounts of whitespace around horizontal rules and are also useful for tabulars outside this package.
Since the `ctable` package imports the `array` and `booktabs` packages, all commands from those packages are available as well.
Note that, in line with the comments that Simon Fear made describing his `booktabs` package, vertical rules for column separation can be produced with `\ctable`, but no provisions are made to have them make contact with horizontal rules.

*This file has version number v1.6b, dated 2004/06/19.

2 Usage

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

```
\ctable[options]       % key=value,...
       {coldefs}       % for \begin{tabular}
       {foottable}     % zero or more \tnote commands (see below)
       {table lines}   % lines 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. Currently the following option keys have been defined:

`caption={...}`	table caption; the braces are needed only if your caption contains a comma.
`cap={...}`	for a short caption to go to the `\tableofcontents`.
`pos=...`	float position, default: `tbp`.
`label=...`	for `\label`
`width=...`	tabularx will be used to typeset the table at the specified width — one or more `X` column specifiers must be provided.
`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.
`center`	center the table in the available text width; this is the default.
`left`	left align the table in the available text width.
`right`	right align the table in the available text width.
`figure`	produce a figure float instead of a table float.
`botcap`	put the caption at the bottom of the float instead of on top of it.
`rotate`	rotate table or figure by 90 degrees anticlockwise and put it on a separate page. If you use this option, the `pos` option is not allowed.
`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.
`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`.
`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).
`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).

The footnotes are placed under the table, without a rule. You therefore probably will want to use the
\tnote `\LL` (last line) command if you use footnotes. `\tnote[label]{footnote text}` places *label* footnote text under the table. Can only be used in the foottable parameter described above. The label is optional, the default label is a single *a*. For more detailed control, you can also replace this command with something like `labeltext&footnotetext\NN`.

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

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

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

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

\ML Middle Line: generates a new line and a thin rule with some extra space over and under it. An optional

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.

dimen parameter sets the line width; the default is 0.05em

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

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

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

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

2.1 The width and maxwidth options

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

3 Examples

3.1 Tables

Table 1 is an example taken from the related package threeparttable.sty 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   = tab:nowidth,
]{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 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 `\LR`, `\MR`, and `\RR` commands of the `booktabs` package, some of the horizontal splitting was regained. By using the `left` option, the table has been left aligned. It was typeset with:

```
\ctable[
   caption = Example with a specified width of 100mm,
   width   = 100mm,
   pos     = b,
   label   = tab:width,
   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
}
```

3.2 Figures

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

```
\ctable[
   caption   = The di- and tri-bromobenzenes,
   label     = fig,
   botcap,
   framebg   = .53 .81 .92,
   framerule = 1pt,
```

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

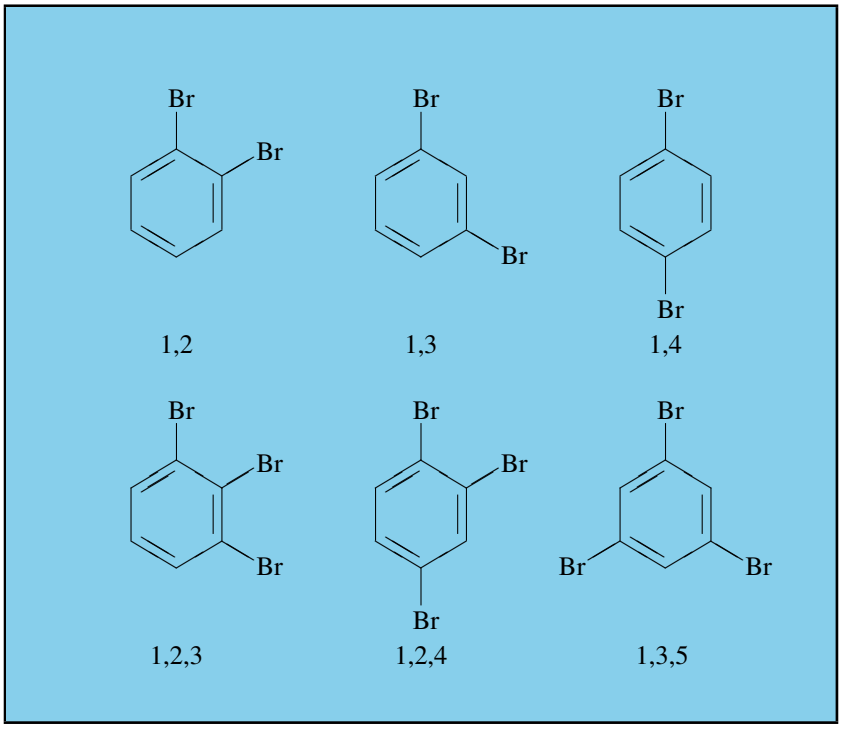

Figure 1: The di- and tri-bromobenzenes

```
   framesep  = 4ex,
   figure,
]{ccc}{}{                       \NN
   \bzdrv{1==Br;2==Br}&
   \bzdrv{1==Br;3==Br}&
   \bzdrv{1==Br;4==Br}          \NN
   1,2 & 1,3 & 1,4              \NN[3ex]
   \bzdrv{1==Br;2==Br;3==Br}&
   \bzdrv{1==Br;2==Br;4==Br}&
   \bzdrv{1==Br;3==Br;5==Br}    \NN
   1,2,3 & 1,2,4 & 1,3,5
}
```

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

4 Implementation

```
⟨*package⟩
\RequirePackage{color,keyval,array,tabularx,booktabs,rotating}
\def\NN{\tabularnewline}
\def\FL{\toprule}
\def\ML{\NN\midrule}
\def\LL{\NN\bottomrule}
\def\@ctblfgcolor#1 #2 #3={\definecolor{@ctblframefg}{rgb}{#1,#2,#3}}
\def\@ctblbgcolor#1 #2 #3={\definecolor{@ctblframebg}{rgb}{#1,#2,#3}}
\newdimen\@ctblframesep
\newdimen\@ctblframerule
\newdimen\@ctblwidth
\newdimen\@ctblmaxwidth
\newdimen\@ctblw % the final width
```

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

```
\newbox\ctbl@tabelx     % the width with X columns
\newbox\ctbl@tabel      % the width where X is replaced with l
\newbox\ctbl@t          % the final box will become one of the two above
```

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

```
\define@key{ctbl}{caption} {\def\@ctblcaption {#1}}
\define@key{ctbl}{cap}     {\def\@ctblcap     {#1}}
\define@key{ctbl}{label}   {\def\@ctbllabel   {#1}}
\define@key{ctbl}{pos}     {\def\@ctblpos     {#1}
                            \def\@ctblbegin{\@ctblbeg[#1]}}
\define@key{ctbl}{width}   {\@ctblwidth       =#1}
\define@key{ctbl}{maxwidth}{\@ctblmaxwidth    =#1}
\define@key{ctbl}{botcap}[]{\def\@ctblbotcap  {1}}
\define@key{ctbl}{rotate}[]{\def\@ctblrotate  {sideways}}
\define@key{ctbl}{figure}[]{\def\@ctbltaborfig{figure}}
\define@key{ctbl}{center}[]{\def\@ctblalign   {center}}
\define@key{ctbl}{right} []{\def\@ctblalign   {flushright}}
\define@key{ctbl}{left}  []{\def\@ctblalign   {flushleft}}
\define@key{ctbl}{star}  []{\def\@ctblstarred {*}}
\define@key{ctbl}{framerule}{\@ctblframerule  =#1}
\define@key{ctbl}{framesep}{\@ctblframesep    =#1}
\define@key{ctbl}{framefg}{\@ctblfgcolor#1=}
\define@key{ctbl}{framebg}{\@ctblbgcolor#1=}
```

a caption will only be generated if the *caption* option was used:

```
\def\@ctblCaption{
   \ifx\@ctblcap\empty\let\@ctblcap\@ctblcaption\fi
   \ifx\@ctblcaption\empty\else
      \caption[\@ctblcap]{\label{\@ctbllabel}\@ctblcaption}
   \fi
}
\newdimen\@ctbloldsep
\newdimen\@ctbloldrule
\def\@ctblframe#1#2#3{%
   \@ctbloldsep\fboxsep\fboxsep\@ctblframesep%
   \@ctbloldrule\fboxrule\fboxrule\@ctblframerule%
   \fcolorbox{#1}{#2}{\fboxsep\@ctbloldsep\fboxrule\@ctbloldrule #3}%
}
\newcommand{\tnote}[2][a]{%
```

```
   \hbox{\@textsuperscript{\normalfont\textit{#1}}}&#2\NN}
\newcommand{\tmark}[1][a]{%
   \hbox{\@textsuperscript{\normalfont\textit{#1}}}}
\newcommand{\ctable}[4][]{
   \def\@ctbltaborfig{table}
   \def\@ctblalign   {center}
   \def\@ctblrotate  {}
   \def\@ctblpos     {}
   \def\@ctblcaption {}
   \def\@ctblcap     {}
   \def\@ctbllabel   {}
   \def\@ctblbeg     {\begin{\@ctblrotate\@ctbltaborfig\@ctblstarred}}
   \def\@ctblbegin   {\@ctblbeg}
   \def\@ctblend     {\end{\@ctblrotate\@ctbltaborfig\@ctblstarred}}
   \def\@ctblbotcap  {}
   \def\@ctblstarred {}
   \definecolor{@ctblframefg}{rgb}{0,0,0}
   \definecolor{@ctblframebg}{rgb}{1,1,1}
   \@ctblframerule0pt
   \@ctblframesep0pt
   \@ctblwidth=0pt
   \@ctblmaxwidth=0pt
   \setkeys{ctbl}{#1}
```

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

```
   \ifdim\@ctblwidth=0pt\else
      \ifdim\@ctblmaxwidth=0pt\else
         \PackageError{ctable}{
            You may not use the width and maxwidth options together}{%
            Use either width or maxwidth}
      \fi
   \fi
   \ifx\@ctblpos\empty\else
      \ifx\@ctblrotate\empty\else
      \PackageError{ctable}{
         You may not use the pos and rotate options together}{%
         Rotated tables and figures are always typeset on a separate page}
      \fi
   \fi
   \newcolumntype{Y}{X}% save the X type
```

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

```
   \sbox\ctbl@tabel{%
      \newcolumntype{X}{l}% temporarily make  type X = l
      \@ctblframe{@ctblframefg}{@ctblframebg}{%
         \begin{tabular}{#2}
            #4
         \end{tabular}%
      }%
   }
```

the other will get the table typeset with the tabularx environment:

```

   \sbox\ctbl@tabelx{%
      \newcolumntype{X}{Y}% restore X
      \@ctblframe{@ctblframefg}{@ctblframebg}{%
         \begin{tabularx}{\ifdim\@ctblwidth>0pt\@ctblwidth\else\@ctblmaxwidth\fi}{#2}
            #4
```

```
      \end{tabularx}%
    }%
  }
```

if no *maxwidth* was given:

```
  \ifdim\@ctblmaxwidth=0pt
```

and also no *width*:

```
    \ifdim\@ctblwidth=0pt
```

then use the tabular environment:

```
      \sbox{\ctbl@t}{\usebox\ctbl@tabel}
    \else
```

if width was given: use the tabularx environment

```
      \sbox{\ctbl@t}{\usebox\ctbl@tabelx}
    \fi
  \else
```

with the *maxwidth* option, we check if the table, typeset with the tabular environment would get too wide:

```
    \ifdim\wd\ctbl@tabel>\@ctblmaxwidth
```

if so, we use the tabularx environment:

```
      \sbox{\ctbl@t}{\usebox\ctbl@tabelx}
    \else
```

but if within limits, we use the tabular environment:

```
      \sbox{\ctbl@t}{\usebox\ctbl@tabel}
    \fi
  \fi
```

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

```
  \@ctblw=\wd\ctbl@t
```

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

```
  \@ctblbegin
    \begin{\@ctblalign}
      \begin{minipage}{\@ctblw}
        \ifx\@ctblbotcap\empty\@ctblCaption\vspace{2ex}\fi
        \usebox\ctbl@t % insert the tabular
        \def\@ctblfootnotes{#3}
        \ifx#3\empty\else % append footnotes, if any
          \\
          \begin{tabularx}{\@ctblw}{r@{\,}>{\footnotesize\raggedright}X}
            #3
          \end{tabularx}
        \fi
        \ifx\@ctblbotcap\empty\else\@ctblCaption\fi
      \end{minipage}
    \end{\@ctblalign}
  \@ctblend
}
⟨/package⟩
```