The \texttt{colortbl} package\footnote{This file has version number \texttt{v1.0h}, last revised 2024/05/26.}

David Carlisle\footnote{Report issues to \url{https://github.com/davidcarlisle/dpctex/issues}}

2024/05/26

\textbf{Abstract}

This package implements a flexible mechanism for giving colored 'panels' behind specified columns in a table. This package requires the \texttt{array} and \texttt{color} packages.

\section{Introduction}

This package is for coloring tables (i.e., giving colored panels behind column entries). In that it has many similarities with Timothy Van Zandt’s \texttt{colortab} package. The internal implementation is quite different though, also \texttt{colortab} works with the table constructs of other formats besides \TeX. This package requires \TeX (and its \texttt{color} and \texttt{array} packages).

First, a standard \texttt{tabular}, for comparison.

\begin{tabular}{|l|c|}
\hline
one & two \\
\hline
three & four \\
\hline
\end{tabular}

\section{The \texttt{\textbackslash columncolor} command}

The examples below demonstrate various possibilities of the \texttt{\textbackslash columncolor} command introduced by this package. The vertical rules specified by | are kept in all the examples, to make the column positioning clearer, although possibly you would not want colored panels and vertical rules in practice.

The package supplies a \texttt{\textbackslash columncolor} command, that should (only) be used in the argument of a > column specifier, to add a colored panel behind the specified column. It can be used in the main ‘preamble’ argument of \texttt{array} or \texttt{tabular}, and also in \texttt{\textbackslash multicol}um\texttt{n} specifiers.

The basic format is:

\texttt{\textbackslash columncolor\{}\langle color model\rangle\}\{\langle color\rangle\} \{\langle left overhang\rangle\} \{\langle right overhang\rangle\}

The first argument (or first two if the optional argument is used) are standard \texttt{color} package arguments, as used by \texttt{\textbackslash color}.

The last two arguments control how far the panel overlaps past the widest entry in the column. If the \texttt{right overhang} argument is omitted then it defaults to
left overhang. If they are both omitted they default to \texttt{tabcolsep} (in \texttt{tabular}) or \texttt{arraycolsep} (in \texttt{array}).

If the overhangs are both set to \texttt{0pt} then the effect is:

\begin{verbatim}
>\{\textcolor{white}{\columncolor[gray]{.2}[0pt]}l
|>{\columncolor[gray]{.8}[0pt]}l
\end{verbatim}

The default overhang of \texttt{tabcolsep} produces:

\begin{verbatim}
>\{\textcolor{white}{\columncolor[gray]{.2}}l
|>{\columncolor[gray]{.8}}l
\end{verbatim}

You might want something between these two extremes. A value of \texttt{.5\tabcolsep} produces the following effect:

\begin{verbatim}
>\{\textcolor{white}{\columncolor[gray]{.2}[.5\tabcolsep]}l
|>{\columncolor[gray]{.8}[.5\tabcolsep]}l
\end{verbatim}

This package should work with most other packages that are compatible with the \texttt{array} package syntax. In particular it works with \texttt{longtable} and \texttt{dcolumn} as the following example shows.

Before starting give a little space: \texttt{\setlength\minrowclearance{2pt}}

\begin{table}[h]
\centering
\begin{tabular}{lp{3cm}l}
\hline
First two columns & Third column D-type (dcolumn) \\
\hline
P-column & p-type and another one & 12.34 \\
\hline
Total & (wrong) & 100.6 \\
\hline
Some long text in the first column & bbb & 1.2 \\
aaa & and some long text in the second column & 1.345 \\
Total & (wrong) & 100.6 \\
\hline
aaa & 1.345 \\
\hline
Note that the colored rules in all columns stretch to accommodate large entries in one column. & bbb & 1.345 \\
\hline
\end{tabular}
\caption{A long table example}
\label{tab:longtable}
\end{table}

Continued...
Depending on your driver you may get unsightly gaps or lines where the ‘screens’ used to produce different shapes interact badly. You may want to cause adjacent panels of the same color by specifying a larger overhang or by adding some negative space (in a \noalign between rows.

<table>
<thead>
<tr>
<th>p-type</th>
<th>D-type (dcolumn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>bbb</td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>aaa</td>
<td>bbb</td>
</tr>
<tr>
<td>12·4</td>
<td></td>
</tr>
<tr>
<td>aaa</td>
<td>bbb</td>
</tr>
<tr>
<td>45·3</td>
<td></td>
</tr>
</tbody>
</table>

The End

This example shows rather poor taste but is quite colorful! Inspect the source file, `colortbl.dtx`, to see the full code for the example, but it uses the following column types.

```latex
\newcolumntype{A}{% >\color{white}\columncolor{red}[.5\tabcolsep]% \raggedright}% p{2cm}
\newcolumntype{B}{% >\columncolor{blue}[.5\tabcolsep]% \color{yellow}\raggedright}% p{3cm}
\newcolumntype{C}{% >\columncolor{yellow}[.5\tabcolsep]% D(.){\cdot}{3.3}}
\newcolumntype{E}{% >{\large\bfseries\columncolor{cyan}[.5\tabcolsep]}c}
\newcolumntype{F}{% >\color{white}\columncolor{magenta}[.5\tabcolsep]}c
\newcolumntype{G}{% >\columncolor{gray}{0.8}[.5\tabcolsep][\tabcolsep]l}
\newcolumntype{H}{% >\columncolor{gray}{0.8}l}
\newcolumntype{I}{% >\columncolor{gray}{0.8}[.5\tabcolsep]l}
```
3 Using the ‘overhang’ arguments for \texttt{tabular*}

The above is all very well for \texttt{tabular}, but what about \texttt{tabular*}?

Here the problem is rather harder. Although \TeX’s \texttt{leader} mechanism which is used by this package to insert the ‘stretchy’ colored panels is rather like \texttt{glue}, the \texttt{tabskip} glue that is inserted between columns of \texttt{tabular*} (and \texttt{longtable} for that matter) has to be ‘real glue’ and not ‘leaders’.

Within limits the overhang options may be used here. Consider the first table example above. If we use \texttt{tabular*} set to 3 cm with a preamble setting of

\begin{verbatim}
\begin{tabular*}{3cm}{%g
@{extracolsep{\fill}}
>{\columncolor[gray]{.8}[0pt][20mm]}l
>{\columncolor[gray]{.8}[5mm][0pt]}l
@{}}
\hline
one & two \\
three & four \\
\end{tabular*}
\end{verbatim}

Changing the specified width to 4 cm works, but don’t push your luck to 5 cm...

\begin{tabular}{|l|c|}
\hline
\rowcolor[gray]{.9} one & two \\
\rowcolor[gray]{.5} three & four \\
\hline
\end{tabular}

4 The \texttt{\rowcolor} command

As demonstrated above, one may change the color of specified rows of a table by the use of \texttt{\multicolumn} commands in each entry of the row. However if your table is to be marked principally by \textit{rows}, you may find this rather inconvenient.

For this reason a new mechanism, \texttt{\rowcolor}, has been introduced\textsuperscript{1}

\texttt{\rowcolor} takes the same argument forms as \texttt{\columncolor}. It must be used at the \textit{start} of a row. If the optional overhang arguments are not used the overhangs will default to the overhangs specified in any \texttt{\columncolor} commands for that column, or \texttt{\tabcolsep/\texttt{arraycolsep} in \texttt{array}}.

If a table entry is in the scope of a \texttt{\columncolor} specified in the table preamble, and also a \texttt{\rowcolor} at the start of the current row, the color specified by \texttt{\rowcolor} will take effect. A \texttt{\multicolumn} command may contain \texttt{\rowcolor}... which will override the default colors for both the current row and column.

\begin{verbatim}
\begin{tabular}{|l|c|}
\hline
\rowcolor[gray]{.9} one & two \\
\rowcolor[gray]{.5} three & four \\
\hline
\end{tabular}
\end{verbatim}

5 The \texttt{\rowcolors} command

The \texttt{\rowcolors} command and its documentation originate in the \texttt{xcolor} package by Dr. Uwe Kern.
\rowcolors \{\ref{commands}\}\{\ref{row}\}\{\ref{odd-row color}\}\{\ref{even-row color}\}
\rowcolors* \{\ref{commands}\}\{\ref{row}\}\{\ref{odd-row color}\}\{\ref{even-row color}\}

One of these commands has to be executed before a table starts. \ref{row} tells the number of the first row which should be colored according to the \ref{odd-row color} and \ref{even-row color} scheme. Each of the color arguments may also be left empty (= no color). In the starred version, \ref{commands} are ignored in rows with inactive \texttt{rowcolors} status (see below), whereas in the non-starred version, \ref{commands} are applied to every row of the table. Such optional commands may be \texttt{\color{red}} or \texttt{\noalign{\null}}.

\showrowcolors \The \texttt{rowcolors} status is activated (i.e., use coloring scheme) by default and/or \texttt{\showrowcolors}. \hiderowcolors, it is inactivated (i.e., ignore coloring scheme) by the command \texttt{\hiderowcolors}. The counter \texttt{\rownum} (or \LaTeX{} counter \texttt{rownum}) may be used within such a table to access the current row number.

At the present time, the \texttt{\rownum} counter is only incremented in tables using \texttt{\rowcolors}.

\rowcolors[\null]{\null}\{\ref{row}\}\{\ref{odd-row color}\}\{\ref{even-row color}\}\arrayrulecolor{\ref{color}}
\begin{tabular}{ll}
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\arrayrulecolor{\ref{color}}
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\rowcolor{\ref{color}}
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\texttt{test} & \texttt{row \therownum} \\
\multicolumn{1}{>{\rowcolor{\ref{color}}}l}{\texttt{test}} & \texttt{row \therownum} \\
\end{tabular}

\section{The \texttt{\cellcolor} command}

A background color can be applied to a single cell of a table by beginning it with \texttt{\cellcolor{\ref{color}}}\{\texttt{test}\}\& \texttt{row \therownum} \},\texttt{\rowcolor{\ref{color}}}\ldots, (or \texttt{\columncolor} if no row-color is in effect) but this has some deficiencies: 1) It prevents data within the cell from triggering the coloration; 2) The alignment specification must be copied from the top of the tabular, which is prone to errors, especially for \texttt{p{}} columns; 3) \texttt{\multicolumn{1}} is just silly. Therefore, there is the \texttt{\cellcolor} command, which works like \texttt{\columncolor} and \texttt{\rowcolor}, but over-rides both of them; \texttt{\cellcolor} can be placed anywhere in the tabular cell to which it applies.

\footnote{At some cost to the internal complexity of this package}
7 Coloring rules.

So you want colored rules as well?

One could do vertical rules without any special commands, just use something like `\textcolor{green}{\vline}` where you'd normally use `. The space between `\vline` will normally be left white. If you want to color that as well, either increase the overhang of the previous column (to `\tabcolsep + \arrayrulewidth + \doublerulesep`) or remove the inter rule glue, and replace by a colored rule of the required thickness. So

```latex
!\textcolor{green}{\vline}
@\textcolor{yellow}{\vrule width \doublerulesep}
!\textcolor{green}{\vline}
```

Should give the same spacing as `\vline` but more color.

However coloring `\hline` and `\cline` is a bit more tricky, so extra commands are provided (which then apply to vertical rules as well).

8 \arrayrulecolor

`\arrayrulecolor` takes the same arguments as `\color`, and is a global declaration which affects all following horizontal and vertical rules in tables. It may be given outside any table, or at the start of a row, or in a > specification in a table preamble. You should note however that if given mid-table it only affects rules that are specified after this point, any vertical rules specified in the preamble will keep their original colors.

9 \doublerulesepcolor

Having colored your rules, you'll probably want something other than white to go in the gaps made by `\vline` or `\hline`. `\doublerulesepcolor` works just the same way as `\arrayrulecolor`. The main thing to note that if this command is used, then `\longtable` will not ‘discard’ the space between `\hline` at a page break. (TEX has a built-in ability to discard space, but the colored ‘space’ which is used once `\doublerulesepcolor` is in effect is really a third rule of a different color to the two outer rules, and rules are rather harder to discard.)

```latex
\setlength{arrayrulewidth}{2pt}\arrayrulecolor{blue}
\setlength{doublerulesepcolor}{2pt}
\begin{tabular}{||l||c||}
\hline\hline
one&two\
three&four\
\hline\hline
\end{tabular}
```

10 More fun with \hhline

The above commands work with `\hhline` from the `hhline` package, however if `hhline` is loaded in addition to this package, a new possibility is added. You may use `>{\ldots}` to add declarations that apply to the following - or = column
rule. In particular you may give \texttt{\arrayrulecolor} and \texttt{\doublerulesepcolor} declarations in this argument.

Most manuals of style warn against over use of rules in tables. I hate to think what they would make of the following rainbow example:

\begin{tabular}{||*7{>{\columncolor{gray}{.9}}c}||}
\rainbowline{t}
\rowcolor[rgb]{.3,.3,1}
Richard\&of\&York\&gave\&battle\&in\&vain\&\multicolumn{1}{>{\columncolor{gray}{.9}}c||}{vain}\\hline
\rainbowline{b}
1\&2\&3\&4\&5\&6\&7\\hline
\end{tabular}

\newcommand\rainbowline[1]{% 
>\{\arrayrulecolor{red}\doublerulesepcolor[rgb]{.3,.3,1}\%
|\#1:=% 
>\{\arrayrulecolor{orange}\doublerulesepcolor[rgb]{.4,.4,1}\%
=\% 
>\{\arrayrulecolor{yellow}\doublerulesepcolor[rgb]{.5,.5,1}\%
=\% 
>\{\arrayrulecolor{green}\doublerulesepcolor[rgb]{.6,.6,1}\%
=\% 
>\{\arrayrulecolor{blue}\doublerulesepcolor[rgb]{.7,.7,1}\%
=\% 
>\{\arrayrulecolor{indigo}\doublerulesepcolor[rgb]{.8,.8,1}\%
=\% 
>\{\arrayrulecolor{violet}\doublerulesepcolor[rgb]{.9,.9,1}\%
=:\#1|\%
\}}
\arrayrulecolor{red}
\doublerulesepcolor[rgb]{.3,.3,1}%
\begin{tabular}{||*7{>{\columncolor{gray}{.9}}c}||}
\rainbowline{t}%
\arrayrulecolor{violet}\doublerulesepcolor[rgb]{.9,.9,1}Richard\&of\&York\&gave\&battle\&in\&\multicolumn{1}{>{\columncolor{gray}{.9}}c||}{vain}\
\rainbowline{b}%
1\&2\&3\&4\&5\&6\&7\end{tabular}

\section{Less fun with \texttt{\cline}}

Lines produced by \texttt{\cline} are colored if you use \texttt{\arrayrulecolor} but you may not notice as they are covered up by any color panels in the following row. This is a ‘feature’ of \texttt{\cline}. If using this package you would probably better using the - rule type in a \texttt{\hhline} argument, rather than \texttt{\cline}. 

7
12 The \texttt{\minrowclearance} command

As this package has to box and measure every entry to figure out how wide to make the rules, I thought I may as well add the following feature. ‘Large’ entries in tables may touch a preceding \texttt{\hline} or the top of a color panel defined by this style. It is best to increase \texttt{\extrarowsep} or \texttt{\arraystretch} sufficiently to ensure this doesn’t happen, as that will keep the line spacing in the table regular. Sometimes however, you just want to \LaTeX{} to insert a bit of extra space above a large entry. You can set the length \texttt{\minrowclearance} to a small value. (The height of a capital letter plus this value should not be greater than the normal height of table rows, else a very uneven table spacing will result.)

Donald Arseneau’s \texttt{tabls} packages provides a similar \texttt{\tablinesep}. I was going to give this the same name for compatibility with \texttt{tabls}, but that is implemented quite differently and probably has different behaviour. So I’ll keep a new name for now.

13 The Code

\begin{verbatim}
1 \langle*package\rangle
Nasty hacky way used by all the graphics packages to include debugging code.
2 \edef\@tempa{\noexpand\AtEndOfPackage{\noexpand\catcode'\noexpand\^^A\the\catcode'\noexpand\^^A\relax}}
3 \@tempa
4 \catcode'\^^A=\catcode'\%
5 \DeclareOption{debugshow}{\catcode'\^^A=9 }
6 All the other options are handled by the color package.
7 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{color}}
8 \ProcessOptions
I need these so load them now. Actually Mark Wooding’s \texttt{mdwtab} package could probably work instead of \texttt{array}, but currently I assume \texttt{array} package internals so...
9 \RequirePackage{array,color}
\@classz First define stub for new \texttt{array} package code.
10 \ifx\do@row@strut\@undefined\let\do@row@strut\relax\fi
\@classz is the main function in the \texttt{array} package handling of primitive column types: It inserts the code for each of the column specifiers, ‘\texttt{clrpmb}’. The other classes deal with the other preamble tokens such as ‘\texttt{c}’ or ‘\texttt{>}’.
11 \def\@classz{$\@classx$
12 \@tempcnta \count@
13 \@tempcnta \count@
14 \prepnext@tok
At this point the color specification for the background panel will be in the code for the ‘\texttt{>}’ specification of this column. This is saved in \texttt{\toks\@emptokena} but \texttt{array} will insert it too late (well it would work for \texttt{c}, but not for \texttt{p}) so fish the color stuff out of that token register by hand, and then insert it around the entry.
Of course this is a terrible hack. What is really needed is a new column type that inserts stuff in the right place (rather like ! but without the spacing that that does). The \texttt{\newcolumntype} command of \texttt{array} only adds ‘second class’
\end{verbatim}
column types. The re-implementations of \texttt{newcolumntype} in my \texttt{blkarray} or Mark Wooding’s \texttt{mdwtab} allow new ‘first class’ column types to be declared, but stick with \texttt{array} for now. This means we have to lift the stuff out of the register before the register gets emptied in the wrong place.

15 \texttt{\textbackslash expandafter\CT@extract\the\toks\@tempcnta\columncolor!\@nil}

Save the entry into a box (using a double group for color safety as usual).

16 \texttt{\@addtopreamble{}}
17 \texttt{\setbox\z@\hbox\bgroup\bgroup}
18 \texttt{\CT@everycr{}}
19 \texttt{\ifcase\@chnum}

\texttt{c} code: This used to use twice as much glue as \texttt{l} and \texttt{r} (1fil on each side). Now modify it to use 1fill total. Also increase the order from 1fil to 1fill to dissuade people from putting stretch glue in table entries.

20 \texttt{\hskip\stretch{.5}\kern\z@}
21 \texttt{\dollartin}
22 \texttt{\insert@column}
23 \texttt{\dollartout\do@row@strut\hskip\stretch{.5}\or}

\texttt{l} and \texttt{r} as before, but using fill glue.

24 \texttt{\dollartin \insert@column \dollartout\do@row@strut \hfill}
25 \texttt{\or}
26 \texttt{\hfill\kern\z@ \dollartin \insert@column \dollartout\do@row@strut}
27 \texttt{\or}

\texttt{m, p and b} as before, but need to take account of array package update.

28 \texttt{\ifx\ar@align@mcell\@undefined}
29 \texttt{$\vcenter$
30 \texttt{\@startpbox{\@nextchar}\insert@column \@endpbox \$}
31 \texttt{\else}
32 \texttt{\@setbox\ar@mcellbox\vbox}
33 \texttt{\@startpbox{\@nextchar}\insert@column \@endpbox}
34 \texttt{\ar@align@mcell}
35 \texttt{\do@row@strut}
36 \texttt{\fi}
37 \texttt{\or}
38 \texttt{\vtop \@startpbox{\@nextchar}\insert@column \@endpbox\do@row@strut}
39 \texttt{\or}
40 \texttt{\vbox \@startpbox{\@nextchar}\insert@column \@endpbox\do@row@strut}
41 \texttt{\fi}

Close the box register assignment.

42 \texttt{\egroup\egroup}

The main new stuff.

43 \texttt{\begingroup}

Initialise color command and overhands.

44 \texttt{\CT@setup}

Run any code resulting from \texttt{\columncolor} commands.

45 \texttt{\CT@columncolor}

Run code from \texttt{\rowcolor} (so this takes precedence over \texttt{\columncolor}).

46 \texttt{\CT@row@color}
Run code from \cellcolor (so this takes precedence over both \columncolor and \rowcolor).

47 \CT@cell@color
This is \relax unless one of the three previous commands has requested a color, in which case it will be \CT@do@color which will insert \leaders of appropriate color.

48 \CT@do@color
49 \endgroup

Nothing to do with color this bit, since we are boxing and measuring the entry anyway may as well check the height, so that large entries don’t bump into horizontal rules (or the top of the color panels).

50 \@tempdim@a\ht\z@  
51 \advance\@tempdim@a\minrowclearance  
52 \vrule\@height\@tempdim@a\@width\z@  

It would be safer to leave this boxed, but unboxing allows some flexibility. However the total glue stretch should either be finite or fill (which will be ignored). There may be fill glue (which will not be ignored) but it should total 0fill. If this box contributes fill glue, then the leaders will not reach the full width of the entry. In the case of \multicolumn entries it is actually possible for this box to contribute shrink glue, in which case the colored panel for that entry will be too wide. Tough luck.

53 \unhbox\z@}  
54 \prepnext@tok

\CT@setup Initialise the overhang lengths and the color command.

55 \def\CT@setup{%  
56 \@tempdim@b\col@sep  
57 \@tempdim@c\col@sep  
58 \def\CT@color{%  
59 \global\let\CT@do@color\CT@do@color\global\let\CT@do@color\CT@do@color  
60 \color)}

\CT@do@color The main point of the package: Add the color panels.

Add a leader of the specified color, with natural width the width of the entry plus the specified overhangs and 1fill stretch. Surround by negative kerns so total natural width is not affected by overhang.

61 \def\CT@do@color{%  
62 \global\let\CT@do@color\relax  
63 \@tempdim@a\wd\z@  
64 \advance\@tempdim@a\@tempdim@b  
65 \advance\@tempdim@a\@tempdim@c  
66 \kern-\@tempdim@b  
67 \leaders\vrule

For quick debugging with xdvi (which can’t do colors). Limit the size of the rule, so I can see the text as well.

68 ^^A  
69 \@height\p@\@depth\p@  
70 \hskip\@tempdim@b\plus 1fill  
71 \kern-\@tempdim@c
Now glue to exactly compensate for the leaders.
\hskip-\wd\z@ \@plus -1fill }

\CT@extract

Now the code to extract the \columncolor commands.
\def\CT@extract#1\columncolor#2#3\@nil{% 
  \if!\noexpand#2% 
    ! is a fake token inserted at the end.
  \else
    If there was an optional argument
    \if\[
      \CT@extractb{#1}#3\@nil
    \else
      No optional argument
      \def\CT@column@color{\CT@color{#2}}%
    \fi
  \fi}
\CT@extractb

Define \CT@column@color to add the right color, and save the overhang lengths. Finally reconstitute the saved ‘>’ tokens, without the color specification. First grab the color spec, with optional arg.
\def\CT@extractb#1#2[ #3]{%
  \def\CT@column@color{\CT@color[ #2]{ #3}}%
} \CT@extractd

Now look for left-overhang (default to \col@sep).
\def\CT@extractd#1{\@testopt{\CT@extracte{#1}}\col@sep}
\CT@extracte

Same for right-overhang (default to left-overhang).
\def\CT@extractf#1[ #2][ #3]#4\columncolor#5\@nil{% 
  \edef\CT@column@color{\CT@column@color \the\@tempdimb\the\@tempdimc\relax}%
  \toks\@tempcnta{#1#4}}%
\CT@everycr

Steal \everypar to initialise row colors
\let\CT@everycr\everycr
\newtoks\everycr
\CT@everycr{\noalign{\global\let\CT@row@color\relax}\the\everycr}
\CT@start
102 \def\CT@start{%
103 \let\CT@arc@save\CT@arc@
104 \let\CT@drsc@save\CT@drsc@
105 \let\CT@row@color@save\CT@row@color
106 \let\CT@cell@color@save\CT@cell@color
107 \global\let\CT@cell@color\relax}
\CT@end
108 \def\CT@end{%
109 \global\let\CT@arc@\CT@arc@save
110 \global\let\CT@drsc@\CT@drsc@save
111 \global\let\CT@row@color\CT@row@color@save
112 \global\let\CT@cell@color\CT@cell@color@save}
\shortstack\shortstack
113 \gdef\@ishortstack#1{%
114 \CT@start\ialign{\mb@l {##}\unskip\mb@r\cr #1\crcr}\CT@end\egroup}
\@tabarray
array and tabular (delayed for delarray)
115 \AtBeginDocument{%
116 \expandafter\def\expandafter\@tabarray\expandafter{%
117 \expandafter\CT@start\@tabarray}}
\endarray
118 \expandafter\def\expandafter\endarray\expandafter{\endarray\CT@end}
\multicolumn
\multicolumn Patch \multicolumn to restore color settings. Done this way to
work wth different versions depending on the age of the array package.
119 \def\@tempa#1\@arstrut#2\relax{%
120 \long\def\multicolumn##1##2##3{%
121 #1%
row@color
122 \let\CT@cell@color\relax
123 \let\CT@column@color\relax
124 \let\CT@do@color\relax
125 \@arstrut
126 #2}}
127 \expandafter\@tempa\multicolumn{#1}{#2}{#3}\relax
128 \let\@temp\relax
\@classvi
Colored rules and rule separations.
129 \def\@classvi{%
\ifcase \@lastchclass
130 \@acol \or
131 \ifx\CT@drsc@\relax
132 \@addtopreamble{\hskip\doublerulesep}%
133 \else
134 \@addtopreamble\{\CT@drsc@\vrule\@width\doublerulesep\}%
135 \fi\or
136 \@acol \or
137 \@classvi
138 \fi}
12
doesn’t really work, as it comes behind the colored panels, but at least make it the right color (the bits you can see, anyway).
\minrowclearance The row height fudge length.
\newlength\minrowclearance
\minrowclearance=0pt

\@mkpream While expanding the \texttt{preamble} \texttt{array} passes tokens through an \texttt{edef}. It doesn’t use \texttt{protection} as it thinks it has full control at that point. As the redefinition above adds \texttt{color}, I need to add that to the list of commands made safe.
\let\@mkpreamarray\@mkpream
\def\@mkpream{\let\CT@setup\relax
\let\CT@color\relax
\let\CT@do@color\relax
\let\color\relax
\let\CT@column@color\relax
\let\CT@row@color\relax
\let\CT@cell@color\relax
\@mkpreamarray}

\CT@do@color For similar reasons, need to make this non-expandable
\let\CT@do@color\relax
\rowcolor
\def\rowcolor{%
\noalign{\ifnum0='}\fi
\global\let\CT@do@color\CT@@do@color
\@ifnextchar[\CT@roa\CT@rob}

\CT@roa
\def\CT@roa[#1]{%\CT@row[\#1]}%\CT@rowc}

\CT@rob
\def\CT@rob#1{%\CT@rowc}

\CT@rowc
\def\CT@rowc{%\CT@rowd{\ifnum'={0}\fi}}

\CT@rowd
\def\CT@rowd[#1]{\@testopt{\CT@rowe[#1]}{#1}}
\CT@rove
\def\CT@rove[#1][#2]{%
\@tempdimb#1%
\@tempdimc#2%
\xdef\CT@row@color{%
\expandafter\noexpand\CT@row@color
\@tempdimb\the\@tempdimb
\@tempdimc\the\@tempdimc
\relax}%
\ifnum0='{\fi}}
\@ifxempty
{⟨arg⟩}{⟨empty⟩}{⟨non-empty⟩}
Tests without expanding, whether the argument {⟨arg⟩} is empty and executes the following code accordingly; {⟨arg⟩} must not start with the token \XC@o. Can also be used within \edef.
\def\@ifxempty#1\@@ifxempty\XC@@{
\ifx#1\@@ifxempty
\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
\rowcolors
\rowcolors*\[
⟨commands⟩\]{⟨row⟩}{⟨odd-row color⟩}{⟨even-row color⟩}
Defines alternating colors for the next tabular environment. Starting with row ⟨row⟩, odd and even rows get their respective colors. The color arguments may also be left empty (= no color). Optional commands may be hline or noalign{⟨stuff⟩}.
In the starred version, ⟨commands⟩ are ignored in rows with inactive rowcolors status (see below), whereas in the non-starred version, ⟨commands⟩ are applied to every row of the table.
\def\rowcolors{
\if@star\@rowcmdfalse\rowc@lors\else\@rowcmdtrue\rowc@lors\fi}
\def\rowc@lors\[#1#2#3#4%{
\global\rownum=\z@
\global\@rowcolorstrue
\@ifxempty{#3}\
{\def\@oddrowcolor{\@norowcolor}}\n{\def\@oddrowcolor{\gdef\CT@row@color{\CT@color{#3}}}}
\@ifxempty{#4}\
{\def\@evenrowcolor{\@norowcolor}}\n{\def\@evenrowcolor{\gdef\CT@row@color{\CT@color{#4}}}}
\if@rowcmd
\def\@rowcolors{#1\if@rowcolors
\noalign{\relax\ifnum\rownum<#2\@norowcolor\else
\ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi\fi%}
\else
\def\@rowcolors{\if@rowcolors
\noalign{\relax\ifnum\rownum<#2\@norowcolor\else
\#1\if@rowcolors
\noalign{\relax\ifnum\rownum<#2\@norowcolor\else
\#1\noalign{\ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi}\fi%}
\fi}%
\else
\def\@rowcolors{\if@rowcolors
\noalign{\relax\ifnum\rownum<#2\@norowcolor\else
\#1\noalign{\ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi}\fi%}
\fi}
241 \fi
242 \CT@everycr{\@rowc@lors\the\everycr}  \\
243 \ignorespaces}
244 \def\@rowc@lors{\noalign{\global\advance\rownum@one\@ne}@\rowcolors}
245 \let\@rowcolors\@empty
246 \showrowcolors
247 \hiderowcolors
248 \if@rowcolors
249 \if@rowcmd
250 \newif\if@rowcolors
251 \newif\if@rowcmd
252 @\ifundefined{rownum}{{\newcounter{rownum}\let\c@rownum\rownum}}{{\let\c@rownum\rownum}
253 \providecommand\therownum{\arabic{rownum}}
254 \edef\cellcolor{\noexpand\protect
255 \expandafter\noexpand\csname cellcolor \endcsname}
256 \@namedef{cellcolor }{%}
257 \@ifnextchar[{{\CT@cellc@firstofone}{\CT@cellc@gobble}[]}%}
258 \AtBeginDocument{%
259 \def\DC@endright{%
260 \expandafter\@tempa{$\hfil\egroup\box\z@\box\tw@}%
261 \ifx\@tempa\DC@endright
262 \newif\if@rowcmd
263 \if@rowcmd
264 \newif\if@rowcolors
265 \edef\cellcolor{\noexpand\protect
266 \expandafter\noexpand\csname cellcolor \endcsname}
267 \@namedef{cellcolor }{%}
268 \@ifnextchar[{{\CT@cellc@firstofone}{\CT@cellc@gobble}[]}%}
269 \AtBeginDocument{%
270 \def\DC@endright{%
271 \def\@tempa{$\hfil\egroup\box\z@\box\tw@}%
272 \if@rowcmd
273 \newif\if@rowcolors
274 \if@rowcolors
275 \if@rowcmd
276 \newif\if@rowcolors
277 \edef\cellcolor{\noexpand\protect
278 \expandafter\noexpand\csname cellcolor \endcsname}
279 \@namedef{cellcolor }{%}
280 \@ifnextchar[{{\CT@cellc@firstofone}{\CT@cellc@gobble}[]}%}
Old dcolumn code.

\AtBeginDocument{%
\ifx\hhline\@undefined\else
\def\HH@box#1#2{\vbox{{\ifx\CT@drsc@\relax\else
\global\dimen\thr@@\tw@\arrayrulewidth
\global\advance\dimen\thr@@\doublerulesep
{\CT@drsc@
\hrule \@height\dimen\thr@@
\vskip-\dimen\thr@@}\fi
\CT@arc@
\hrule \@height\arrayrulewidth \@width #1
\vskip\doublerulesep
\hrule \@height\arrayrulewidth \@width #2}}}
\def\HH@loop{\ifx\@tempb\'\def\next##1{\the\toks@\cr}\else\let\next\HH@let
\ifx\@tempb|\if@tempswa
\ifx\CT@drsc@\relax
\HH@add{{\hskip\doublerulesep}\CT@arc@\vrule\@width\doublerulesep}\else
\HH@add{{\CT@drsc@\vrule\@width\doublerulesep}}\fi
\@tempswatrue
\HH@add{{\CT@arc@\vline}}\else
\ifx\@tempb:\if@tempswa
\HH@add{{\hskip\doublerulesep}}\fi\@tempswatrue
\HH@add{{\CT@arc@\vline\copy\@ne\@tempc\vline}}\else
\ifx\@tempb~\@tempswafalse
\if@firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswatrue
\HH@add{{\CT@arc@\vline}}\else
\HH@add{{\CT@arc@\vline\copy\@one@\@tempc\vline}}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc\HH@box\arrayrulewidth\arrayrulewidth\@tempc}\else
\HH@add{\@tempc\HH@box\arrayrulewidth\arrayrulewidth\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi\@tempswartue
\HH@add{\@tempc}\else
\HH@add{\@tempc}\else
\if\firstamp\@firstampfalse\else\HH@add{&\omit}\fi
\fi}}
Stop the backspacing for `t` and `b`, it messes up the underlying color.

\textbackslash AtBeginDocument{
    \textbackslash if\textbackslash x\textbackslash CT\textbackslash 0\textbackslash drsc\textbackslash 0\relax
    \textbackslash HH\textbackslash add\textbackslash (\textbackslash hfil\textbackslash )\textbackslash else
    \textbackslash HH\textbackslash add\textbackslash (\%\textbackslash CT\textbackslash 0\textbackslash drsc\textbackslash 0\leaders\textbackslash hrule\textbackslash @height\textbackslash HH\textbackslash 0\textbackslash height\textbackslash hfil\textbackslash )\textbackslash f\textbackslash i
    \textbackslash fi
}

\textbackslash if\textbackslash x\textbackslash @tempb\textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbackslash \textbacklash}
\else
  \global\let\LT@next\empty
  \gdef\CT@LT@sep{%
    \noalign{\penalty-\@lowpenalty\vskip-\arrayrulewidth}%
    \fi
  \ifnum0='\fi%
  \multispan\LT@cols
  {\CT@arc@leaders\hfill\CT@LT@sep}
  \multispan\LT@cols
  {\CT@arc@leaders\hfill\CT@LT@sep}
  \noalign{\penalty\@M}%
  \LT@next}\
\fi
\end{package}