The \texttt{hhline} package

David Carlisle

2020/01/04

Abstract

\texttt{\textbackslash hhline} produces a line like \texttt{\textbackslash hline}, or a double line like \texttt{\textbackslash hline\textbackslash hline}, except for its interaction with vertical lines.

1 Introduction

The argument to \texttt{\textbackslash hhline} is similar to the preamble of an \texttt{array} or \texttt{tabular}. It consists of a list of tokens with the following meanings:

= A double hline the width of a column.
- A single hline the width of a column.
~ A column with no hline.
| A vline which ‘cuts’ through a double (or single) hline.
: A vline which is broken by a double hline.
# A double hline segment between two vlines.
t The top half of a double hline segment.
b The bottom half of a double hline segment.
* *\{3\}{==#} expands to ==#==#==#, as in the *-form for the preamble.

If a double vline is specified (|| or ::) then the hlines produced by \texttt{\textbackslash hhline} are broken. To obtain the effect of an hline ‘cutting through’ the double vline, use a # or omit the vline specifiers, depending on whether or not you wish the double vline to break.

The tokens \texttt{t} and \texttt{b} must be used between two vertical rules. \texttt{|tb|} produces the same lines as #, but is much less efficient. The main use for these are to make constructions like \texttt{\textbackslash t:} (top left corner) and \texttt{:b|} (bottom right corner).

If \texttt{\textbackslash hhline} is used to make a single hline, then the argument should only contain the tokens -, ~ and | (and *-expressions).

\footnotesize

\footnotesize{This file has version number v2.04, last revised 2020/01/04.}
An example using most of these features is:

\begin{tabular}{||cc||c|c||}
\hline
a&b&c&d\
\hline
1&2&3&4\
\hline
i&j&k&l\
\hline
w&x&y&z\
\hline
\end{tabular}

The lines produced by \LaTeX’s \texttt{\hline} consist of a single (T\EX primitive) \texttt{\hrule}. The lines produced by \texttt{\hhline} are made up of lots of small line segments. \LaTeX will place these very accurately in the \texttt{.dvi} file, but the program that you use to print the \texttt{.dvi} file may not line up these segments exactly. (A similar problem can occur with diagonal lines in the \texttt{picture} environment.)

If this effect causes a problem, you could try a different driver program, or if this is not possible, increasing \texttt{\arrayrulewidth} may help to reduce the effect.

## 2 The Macros

\HH@box Makes a box containing a double hline segment. The most common case, both rules of length \texttt{\doublerulesep} will be stored in \texttt{\box1}. this is not initialised until \texttt{\hhline} is called as the user may change the parameters \texttt{\doublerulesep} and \texttt{\arrayrulewidth}. The two arguments to \texttt{\HH@box} are the widths (ie lengths) of the top and bottom rules.

\HH@add Build up the preamble in the register \texttt{\toks0}.

\HH@xexpast We ‘borrow’ the version of \texttt{\@xexpast} from Mittelbach’s array.sty, as this allows # to appear in the argument list.

\HH@xexpast1 We ‘borrow’ the version of \texttt{\@xexpast} from Mittelbach’s array.sty, as this allows # to appear in the argument list.
\hline Use a simplified version of \@mkpream to break apart the argument to \hhline. Actually it is oversimplified, It assumes that the vertical rules are at the end of the column. If you were to specify c|@{xx}| in the array argument, then \hhline would not be able to access the first vertical rule. (It ought to have an @ option, and add \leaders up to the width of a box containing the @-expression. We use a loop made with \futurelet rather than \@tfor so that we can use # to denote the crossing of a double hline with a double vline. \if@firstamp is true in the first column and false otherwise. \if@tempswa is true if the previous entry was a vline (:, | or #).

\def\hhline#1{% Put two rules of width \doublerulesep in \box1
\global\setbox\@ne\HH@box\doublerulesep\doublerulesep
If Mittelbach’s array.sty is loaded, we do not need the negative \hskip’s around vertical rules.
\def\HH@xexnoop#1\@@{}% Now expand the *-forms and add dummy tokens ( \relax and’) to either end of the token list. Call \HH@let to start processing the token list.
\xdef\@tempc{\ifx\extrarowheight\HH@undef\hskip-.5\arrayrulewidth\fi}%% \HH@xexpast\relax#1*0x\@@\toks@{}\expandafter\HH@let\@tempa'}

\HH@let Discard the last token, look at the next one.
\def\HH@let#1{\futurelet\@tempb\HH@loop}

\HH@loop The main loop. Note we use \ifx rather than \if in version 2 as the new token ~ is active.
\def\HH@loop{% If next token is ', stop the loop and put the lines into this row of the alignment.
\ifx\@tempb'\def\next##1{\the\toks@\cr}\else\let\next\HH@let
|, add a vertical rule (across either a double or single hline).
\ifx\@tempb|\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
\HH@add{\@tempc\vline\@tempc}\else
:, add a broken vertical rule (across a double hline).
\ifx\@tempb:\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
\HH@add{\@tempc\vline\@tempc}\else
#, add a double hline segment between two vlines.
\ifx\@tempb#\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
\HH@add{\@tempc\HH@box\arrayrulewidth\arrayrulewidth\@tempc}\else
~, A column with no hline (this gives an effect similar to \cline).
\ifx\@tempb~\@tempswafalse
\HH@add{\&\omit}\fi\HH@add{\hfil}\else...
-, add a single hline across the column.
\\ifx@tempb-@tempswafalse
  \\if@firstamp@firstampfalse\else\HH@add{&\omit}\fi
  \\HH@add{\leaders\hrule@height\arrayrulewidth\hfil}\else
=, add a double hline across the column.
\\ifx@tempb=@tempswafalse
  \\if@firstamp@firstampfalse\else\HH@add{&\omit}\fi
Put in as many copies of \box1 as possible with \leaders, this may leave gaps at the ends, so put an extra box at each end, overlapping the \leaders.
\HH@add
\\ifx@tempb\@ne\HH@add{%rlap{\copy@ne}\leaders\copy@one\hfil\llap{\copy@one}}\else
t, add the top half of a double hline segment, in a \rlap so that it may be used with b.
\HH@add
\\ifx@tempb t\HH@add{%rlap{\HH@box\doublerulesep\z@}}\else
b, add the bottom half of a double hline segment in a \rlap so that it may be used with t.
\HH@add
\\ifx@tempb b\HH@add{%rlap{\HH@box\z@\doublerulesep}}\else
space, Gobble the space and loop again.
\HH@add
\\ifx@tempb@stoken\let\next\HH@spacelet\else
Otherwise ignore the token, with a warning.
\PackageWarning{hhline}\
\\meaning@tempb\space ignored in \noexpand\hhline argument\MessageBreak\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
\next
\\HH@spacelet Helper macro to gobble a space token and continue the loop.
\lowercase{\def\HH@spacelet}{\futurelet@tempb \HH@loop}
\lowercase{\def\HH@loop}{\ifx@tempb=\futurelet@tempb\HH@spacelet\else\fi\fi\fi\fi\fi\fi\fi\fi\fi
"