The \texttt{hhline} package\textsuperscript{*}

David Carlisle
carlisle@cs.man.ac.uk
1994/05/23

Abstract
\texttt{\hline} produces a line like \texttt{\hline}, or a double line like \texttt{\hline\hline},
exCEPT for its interaction with vertical lines.

1 Introduction

The argument to \texttt{\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 \texttt{==#==#==#}, as in the \*-form for the preamble.

If a double vline is specified (|| or ::) then the hlines produced by \texttt{\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 \texttt{#}, but is much less efficient. The main use for these are to make
constructions like \texttt{|t:} (top left corner) and \texttt{:b|} (bottom right corner).

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

An example using most of these features is:

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

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

*This file has version number v2.03, last revised 1994/05/23.
The lines produced by \LaTeX{}’s \texttt{\textbackslash hline} consist of a single (\TeX{} primitive) \texttt{\textbackslash hrule}. The lines produced by \texttt{\textbackslash hhline} are made up of lots of small line segments. \TeX{} 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{\textbackslash arrayrulewidth} may help to reduce the effect.

## 2 The Macros

\begin{verbatim}
\HH@box Makes a box containing a double hline segment. The most common case, both rules of length \texttt{\textbackslash doublerulesep} will be stored in \texttt{\box1}, this is not initialised until \texttt{\textbackslash hline} is called as the user may change the parameters \texttt{\textbackslash doublerulesep} and \texttt{\textbackslash 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{\textbackslash @xexpast} from Mittelbach’s \texttt{array.sty}, as this allows # to appear in the argument list.

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

\hhline Put two rules of width \texttt{\doublerulesep} in \texttt{\box1}.

If Mittelbach’s \texttt{array.sty} is loaded, we do not need the negative \texttt{\hskip}’s around vertical rules.
\end{verbatim}

2
Now expand the \* -forms and add dummy tokens (\texttt{relax} and `) to either end of the token list. Call \texttt{\@let} to start processing the token list.

\begin{verbatim}
\HH@let Discard the last token, look at the next one.
\HH@loop The main loop. Note we use \texttt{\ifx} rather than \texttt{\if} in version 2 as the new token ` is active.
\end{verbatim}