The hhline package*

David Carlisle
carlisle@cs.man.ac.uk

1994/05/23

Abstract

`\hhline` produces a line like `\hline`, or a double line like `\hline\hline`, except for its interaction with vertical lines.

1 Introduction

The argument to `\hhline` is similar to the preamble of an `array` or `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 `\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 `t` and `b` must be used between two vertical rules. `|tb|` produces the same lines as `#`, but is much less efficient. The main use for these are to make constructions like `|t:` (top left corner) and `:b|` (bottom right corner).

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

*This file has version number v2.03, last revised 1994/05/23.

An example using most of these features is:

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

a	b	c	d
1	2	3	4
i	j	k	l
w	x	y	z

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

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