

# L'extension `hhline`\*

David Carlisle

28/10/2014

Ce fichier est maintenu par l'équipe du «`LATEX` Project». Les rapports d'anomalie peuvent être envoyés en anglais à <http://latex-project.org/bugs.html> (catégorie `tools`).

## Résumé

Cette extension augmente les possibilités de filets horizontaux et verticaux dans un environnement `array` ou `tabular`. Il introduit la nouvelle commande `\hhline` qui produit un filet comme `\hline` ou un double filet comme `\hline\hline`, sauf pour leur interaction avec les filets verticaux.

## 1 Introduction

L'argument de `\hhline` est similaire au préambule d'un environnement `array` ou `tabular`. Il consiste en une liste d'unités lexicales (*tokens*) ayant les significations suivantes :

- = Un filet horizontal double sur la largeur d'une colonne.
- Un filet horizontal simple sur la largeur d'une colonne.
- ~ Une colonne sans filet horizontal.
- | Un filet vertical qui croise un filet horizontal simple ou double.
- : Un filet vertical qui est brisé par un filet horizontal double.
- # Un filet horizontal double segment entre deux filets verticaux.
- t La moitié supérieure d'un filet horizontal double.
- b La moitié inférieure d'un filet horizontal double.
- \* `*{3}{==#}` se développe en `==#==#==#`, comme avec `*` dans le préambule.

Si un double filet vertical est spécifié (`|` ou `:`) alors les filets horizontaux produits par `\hhline` sont brisés. Pour obtenir l'effet d'un filet horizontal « passant au

---

\*Ce fichier a pour numéro de version v2.03 et a été mis à jour le 28/10/2014. La première traduction, basée sur la version v2.03, a été publiée par Jean-Pierre Drucbert en 2000.

travers » d'un filet vertical double, utilisez un `#` ou omettez les spécificateurs de filets verticaux, selon que vous vouliez que le double filet vertical soit brisé ou pas.

Les unités lexicales `t` et `b` doivent être utilisées entre deux filets verticaux. La combinaison `|tb|` produit les mêmes lignes que `#` mais est moins efficace. Ces unités lexicales sont plutôt pensées pour faire des combinaisons telles que `|t:` (coin supérieur gauche) et `:b|` (coin inférieur droit).

Si `\hline` est utilisée pour faire un simple filet horizontal, alors son argument ne doit contenir que des unités lexicales `-`, `~` et `|` (et des expressions débutant par `*`).

Voici un exemple utilisant la plupart de ces possibilités :

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

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

Les filets produits par la commande `\hline` de  $\text{\LaTeX}$  sont formées par une seule primitive de  $\text{\TeX}$ , `\hrule`. Les filets produits par `\hline` sont constitués de nombreux petits segments.  $\text{\TeX}$  les place très précisément dans le fichier `.dvi`, mais le programme que vous utilisez pour imprimer le fichier `.dvi` peut ne pas les aligner exactement (un problème similaire peut survenir avec les lignes obliques dans l'environnement `picture`).

Si ceci pose un problème, vous pouvez soit essayer un autre programme d'impression (si cela est possible), soit augmenter la dimension `\arrayrulewidth` pour essayer de réduire les effets.

## 2 Les commandes

1 `\package`

`\HH@box` Construit une boîte contenant un segment de double filet horizontal. Dans le cas le plus courant, deux filets de longueur `\doublerulesep` seront stockés dans la boîte `\box1`. Ceci n'est pas initialisé tant que `\hline` n'est pas appelé car l'utilisateur peut changer les paramètres `\doublerulesep` et `\arrayrulewidth`. Les deux arguments de `\HH@box` sont les largeurs (autrement dit les longueurs) du filet du haut et du filet du bas.

```
2 \def\HH@box#1#2{\vbox{%
3 \hrule \@height \arrayrulewidth \@width #1
```

```

4 \vskip \doublerulesep
5 \hrule \@height \arrayrulewidth \@width #2}}

\HH@add Développe le préambule dans le registre \toks@.
6 \def\HH@add#1{\toks@\expandafter{\the\toks@#1}}

\HH@xexpast Nous « empruntons » la version de \xexpast de l'extension de Mittelbach, car
\HH@xexnoop ceci permet de faire apparaître # dans la liste des arguments.
7 \def\HH@xexpast#1*#2#3#4\@@{%
8   \@tempcnta #2
9   \toks@={#1}\temptokena={#3}%
10  \let\the@toksz\relax \let\the@toks\relax
11  \def\@tempa{\the@toksz}%
12  \ifnum\@tempcnta >0 \@whilenum\@tempcnta >0\do
13    {\edef\@tempa{\@tempa\the@toks}\advance \@tempcnta \m@ne}%
14    \let \@tempb \HH@xexpast \else
15    \let \@tempb \HH@xexnoop \fi
16  \def\the@toksz{\the\toks@}\def\the@toks{\the\temptokena}%
17  \edef\@tempa{\@tempa}%
18  \expandafter \@tempb \@tempa #4\@@}
19
20 \def\HH@xexnoop#1\@@{}

✖

\hhline La commande utilise une version simplifiée de \mkpream pour décomposer l'ar-
gument à destination de \hhline. En fait, elle est quasiment schématique. Elle
suppose que les filets verticaux sont en fin de colonne. Si vous aviez à spécifier
c|@{xx}| dans l'argument du tableau, alors \hhline ne serait pas capable d'ac-
céder au premier filet vertica. (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 #).
21 \def\hhline#1{\omit\@firstamptrue\@tempswafalse
Put two rules of width \doublerulesep in \box1
22 \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.
23 \xdef\@tempc{\ifx\extrarowheight\HH@undef\hskip-.5\arrayrulewidth\fi}%
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.
24 \HH@xexpast\relax#1*0x\@@\toks@{}\expandafter\HH@let\@tempa'}

\HH@let Discard the last token, look at the next one.
25 \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.
26 \def\HH@loop{%
If next token is ‘, stop the loop and put the lines into this row of the alignment.
27 \ifx\@tempb'\def\next##1{\the\toks@\cr}\else\let\next\HH@let
|, add a vertical rule (across either a double or single hline).
28 \ifx\@tempb|\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
29 \HH@add{\@tempc\vline\@tempc}\else
:, add a broken vertical rule (across a double hline).
30 \ifx\@tempb:\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
31 \HH@add{\@tempc\HH@box\arrayrulewidth\arrayrulewidth\@tempc}\else
#, add a double hline segment between two vlins.
32 \ifx\@tempb##\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
33 \HH@add{\@tempc\vline\@tempc\copy\@ne\@tempc\vline\@tempc}\else
~, A column with no hline (this gives an effect similar to \cline).
34 \ifx\@tempb~\@tempswafalse
35 \if@firststamp\@firststampfalse\else\HH@add{&\omit}\fi
36 \HH@add{\hfil}\else
-, ajout d'un filet horizontal simple à travers de la colonne.
37 \ifx\@tempb-\@tempswafalse
38 \if@firststamp\@firststampfalse\else\HH@add{&\omit}\fi
39 \HH@add{\leaders\hrule\@height\arrayrulewidth\hfil}\else
=, ajout d'un filet horizontal double à travers de la colonne.
40 \ifx\@tempb=\@tempswafalse
41 \if@firststamp\@firststampfalse\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. ✖
42 \HH@add
43 {\rlap{\copy\@ne}\leaders\copy\@ne\hfil\llap{\copy\@ne}}\else
t, ajoute la moitié haute d'un segment de filet horizontal double dans une com-
mande \rlap afin qu'elle puisse être utilisée avec b.
44 \ifx\@tempb t\HH@add{\rlap{\HH@box\doublerulesep\z@}}\else
b, ajoute la moitié basse d'un segment de filet horizontal double dans une com-
mande \rlap afin qu'elle puisse être utilisée avec t.
45 \ifx\@tempb b\HH@add{\rlap{\HH@box\z@\doublerulesep}}\else
Sinon, l'unité lexicale est ignorée, avec un avertissement l'indiquant.
46 \PackageWarning{hhline}%
47 {\meaning\@tempb\space ignored in \noexpand\hhline argument%
48 \MessageBreak}%
49 \fi\fi\fi\fi\fi\fi\fi\fi\fi
Puis la boucle est parcourue à nouveau.
50 \next}

```

51 `</package>`