The \texttt{tabularht} package

Heiko Oberdiek\(*\)

2019/12/29 v2.7

Abstract

This package defines some environments that adds a height specification
to tabular and array.

Contents

1 Usage  
  1.1 Option \texttt{vlines} ................................. 2  
  1.2 Limitations ........................................ 3  
  1.3 Compatibility ..................................... 3  
  1.4 Examples .......................................... 3  
    1.4.1 Example 1 ................................... 3  
    1.4.2 Example 2 ................................... 4  

2 Implementation  
  2.1 Environments ...................................... 4  
  2.2 Options ........................................... 7  
  2.3 Option \texttt{vlines}, driver independent stuff ....... 7  
  2.4 Driver \texttt{pdftex} ................................. 8  
  2.5 DVI drivers ...................................... 11  

3 Installation  
  3.1 Download .......................................... 14  
  3.2 Bundle installation ............................... 14  
  3.3 Package installation .............................. 14  
  3.4 Refresh file name databases ..................... 15  
  3.5 Some details for the interested ................. 15  

4 History  
[2005/09/22 v1.0] ....................................... 15  
[2005/10/16 v2.0] ....................................... 15  
[2005/10/18 v2.1] ....................................... 16  
[2006/02/20 v2.2] ....................................... 16  
[2006/12/22 v2.3] ....................................... 16  
[2007/03/21 v2.4] ....................................... 16  
[2007/04/11 v2.5] ....................................... 16  
[2016/05/16 v2.6] ....................................... 16  
[2019/12/29 v2.7] ....................................... 16  

5 Index  

\footnotesize\(*\)Please report any issues at \url{https://github.com/ho-tex/oberdiek/issues}
1 Usage

\usepackage{tabularht}

The package provides the following environments that extend the tabular/array environment by a height specification as first argument:

- \texttt{tabularht}, \texttt{tabularht*}
- \texttt{arrayht}
- \texttt{tabularhtx} (if package \texttt{tabularx} is loaded)

The height argument allows a length specification, package \texttt{calc} is supported if used. This means, the tabular will have the specified height. You can also use the prefixes \texttt{to=} and \texttt{spread=}. \texttt{to=} is the default, \texttt{spread=} means, the natural height of the tabular box is changed by the length after \texttt{spread=}.

Examples:

\begin{tabularht}{1in} → height is 1in
\begin{tabularht}{to=1in} → height is 1in
\begin{tabularht}{spread=0pt} → natural height, same as \begin{tabular
\begin{tabularht}{spread=1in} → natural height increased by 1in

Hint: See also package \texttt{tabularkv}, it provides an interface, where most parameters for the environments can be given by key-value pairs.

\begin{verbatim}
\interrowspace{...}
\end{verbatim}

Adds space between table rows. It is essentially the same as \texttt{\noalign{\vspace{...}}}. 

\begin{verbatim}
\interrowfill
\end{verbatim}

Short for \texttt{\interrowspace{\fill}}

\begin{verbatim}
\interrowstart...\interrowstop
\end{verbatim}

Marker commands, useful for option \texttt{vlines}.

1.1 Option \texttt{vlines}

Warning: This stuff is experimental.

Vertical lines are interrupted, if space is inserted in \texttt{\noalign}, \texttt{\interrowspace}, \texttt{\addlinespace} (\texttt{booktabs}), between double \texttt{\hlines}. This option tries to detect and add the vertical lines. The lines in a tabular with \texttt{tabularht} support (environments of this package) are numbered from left to right. The gap that is controlled by \texttt{\interrowspace} or inbetween \texttt{\interrowstart} and \texttt{\interrowstop} is then filled with the detected vertical lines.

If only a limited selection of the lines should be drawn, the commands know an optional argument with a list of line numbers, e.g.

\begin{verbatim}
\begin{tabularht}{50mm}{|l|l|}
Hello & World\\
\interrowfill[1,3] Foo & Bar
\end{tabularht}
\end{verbatim}
There are three lines, but the middle line is not drawn in the gap between the first and second row. Zero can be used to suppress all lines:

\interrowspace[0]{10mm}

The syntax of the commands with the optional argument with the line number list \((list)\). \((list)\) is a comma separated list of numbers, \((height)\) means the height specification described above with the optional prefixes \(to\)= or \(spread\)=.

\interrowspace [\((list)\)] \{(height)\}
\interrowfill [\((list)\)]
\interrowstart [\((list)\) ... \interrowstop

Option \texttt{vlines} is driver dependent and uses \(\varepsilon\)-\TeX features.

\textbf{pdftex}: \texttt{pdf\TeX} in PDF mode. Here the positions of the lines are written with the help of the \texttt{pdfsavepos} feature into the .\texttt{aux} file(s). Therefore you need two \LaTeX runs to get the lines.

\textbf{dvips}: Here, PostScript's currentpoint it used to get the line positions. The lines are then drawn at the end of the page. Thus one \LaTeX/dvips run is sufficient for this option.

\textbf{Other drivers}:

\textbf{PostScript drivers}: probably possible, an end of page hook would be nice.
\textbf{VT\TeX}: with GeX (PostScript interpreter) probably possible.
\textbf{dvipdfm}: no idea. The big problem is, how to get the current position?

1.2 Limitations

- Vertical lines are interrupted by \texttt{\noalign{\vfill}}.

1.3 Compatibility

- \texttt{array, delarray, tabularx} are supported.
- There can be problems with packages that redefine \texttt{\@array} (or \texttt{\@\@array}, \texttt{\@tabarray}) and \texttt{\@arrayrule} (for option \texttt{vlines}).
- \texttt{colortbl}: it should at least work, but there isn’t support for filling the gaps with color, neither the rules nor the backgrounds.

1.4 Examples

1.4.1 Example 1

1 \{\texttt{\*example1}\}
2 \documentclass\{article\}
3 \usepackage\{tabularht\}
4
5 \begin\{document\}
6 \fbox\{%
7 \begin\{tabularht\}\{1in\}\{4in\}\{@{}l@{\extracolsep\{\fill\}}r@{}}\%
8 upper left corner & upper right corner\%
9 \noalign{\vfill} upper left corner & upper right corner \%
10 \noalign{\vfill} upper left corner & upper right corner \%
11 \noalign{\vfill}
12 \begin{tabularht*}%
13 \end{tabularht*}%
14 \end{document}
15 /*example1*/

1.4.2 Example 2

(*example2)
\documentclass{article}
\usepackage{booktabs}
\usepackage[dvips,vlines]{tabularht}
\begin{document}
\begin{tabularht}{spread=0pt}{|l|l|}
\hline
First&Line\%
\hline
\interrowstart
\addlinespace[10mm]%
\interrowstop
\hline
Second&Line\%
\interrowstart
\hline
Third&Line\%
\interrowspace{10mm}
\hline
Fourth&Line\%
\hline
\end{tabularht}
\end{document}
/*example2*/

2 Implementation

(*package)

Package identification.
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{tabularht}%
[2019/12/29 v2.7 Tabular with height specified (HO)]

2.1 Environments

\let@toarrayheight\@empty
\let@tabH@array@init\@empty
\toks@={%
\begingroup
\long\def\x#1\vcenter{fi}\bf\bgroup\sharp\sharp\sharp\sharp\nil{%
\endgroup
\gdef@array[#1]##2{%
\tabH@array@init
#1%
\vcenter{fi}\fi
\@toarrayheight
bgroup
\let\@toarrayheight\@empty
#2\sharp###3#4%
}
\expandafter\x\@array[\#1]{\#2}\@nil % hash-ok
\edef	abH@patch@array{\the\toks@}
\def	abH@patch@@array{%
\ifx\@array\@@array
\def\reserved@a{\let\@@array\@array}%
\else
\let\reserved@a\relax
\fi
\tabH@patch@array
\reserved@a
\tabH@patch@@array
\@ifpackageloaded{array}{}{%
\AtBeginDocument{%
\@ifpackageloaded{array}{{
\tabH@patch@@array
}{%}
}
}%
\def	abH@setheight#1{%
\tabH@@setheight#1==\@nil
}
\def	abH@@setheight#1=#2=#3\@nil{%
\ifx\#2\#3\%
\setlength{\dimen@}{#1}%
\edef\@toarrayheight{to\the\dimen@}%
\else
\edef	abH@temp{\zap@space#1 \@empty}%
\ifx\tabH@temp\tabH@to
\else
\ifx\tabH@temp\tabH@spread
\else
\PackageError{tabularht}{Unknown height specifier '{\expandafter\strip@prefix\meaning\tabH@temp'}%}
\MessageBreak
The height dimension for tabular height can be prefixed with 'to=' or 'spread=', default is 'to='.%
\let\tabH@temp\tabH@to
\fi
\fi
\fi
\setlength{\dimen@}{#2}%
\edef\@toarrayheight{\tabH@temp\the\dimen@}%
\fi
\fi
\setlength{\dimen@}{#2}%
\edef\@toarrayheight{\tabH@temp\the\dimen@}%
\fi
\fi
\def\tabH@to{to}
\def\tabH@spread{spread}
First argument is the height of the table, then the original arguments for tabular follow.
\newenvironment{tabularht}{%}
\tabH@setheight{#1}%
\tabular%
\endtabular%
}%
\newenvironment{tabularht*}{%}
\tabH@setheight{#1}%
@nameuse{tabular*}%
}%
@nameuse{endtabular*}%
}%
\newenvironment{tabularhtx}{%}
\tabH@setheight{#1}%
\tabularx%
\endtabularx%
}%
\newenvironment{arrayht}{%}
\tabH@setheight{#1}%
\array%
\endarray%
}%
def\interrowspace{%
\noalign\bgroup
\tabH@interrowspace
\egroup
}%
\newcommand*{\tabH@interrowspace}[2]{%\tabH@vspace{#1}{#2}\
\egroup
}%
def\interrowfill{%
\noalign\bgroup
\tabH@interrowfill
\egroup
}%
\newcommand*{\tabH@interrowfill}[1]{%\tabH@vspace{#1}{\fill}\
\egroup
}%
\def\tabH@vspace#1#2{\tabH@vspace@start{#1}vspace{#2}\tabH@vspace@stop}
\let\tabH@vspace@start\@gobble
\let\tabH@vspace@stop\@empty
\newcommand*{\interrowstart}{\noalign\bgroup\tabH@interrowstart\egroup}%
\def\interrowfill{%\noalign\bgroup\tabH@interrowfill\egroup}%
\newcommand*{\tabH@interrowfill}[1]{%\tabH@vspace{#1}{\fill}\
\egroup}%
\def\tabH@vspace#1#2{\tabH@vspace@start{#1}vspace{#2}\tabH@vspace@stop}
\let\tabH@vspace@start\@gobble
\let\tabH@vspace@stop\@empty
\newcommand*{\interrowstart}{\noalign\bgroup\tabH@interrowstart\egroup}%
\def\interrowfill{%\noalign\bgroup\tabH@interrowfill\egroup}%
\newcommand*{\tabH@interrowfill}[1]{%\tabH@vspace{#1}{\fill}\
\egroup}%
\def\tabH@vspace#1#2{\tabH@vspace@start{#1}vspace{#2}\tabH@vspace@stop}
\let\tabH@vspace@start\@gobble
\let\tabH@vspace@stop\@empty
\newcommand*{\interrowstart}{\noalign\bgroup\tabH@interrowstart\egroup}
\newcommand*{\tabH@interrowstart}[1]{% 
\tabH@vspace@start{#1}\
\egroup
}

\newcommand*{\interrowstop}{%
\noalign{\tabH@vspace@stop}%
}

2.2 Options

\providecommand*{\tabH@driver}{}

\DeclareOption{vlines}{%
\let\tabH@temp\relax
%
}{}
\DeclareOption{pdftex}{%
\DeclareOption{dvips}{%
\def\tabH@driver{dvips}%
%
}}{}
\ProcessOptions*\relax

\ifx\tabH@temp\relax
\else
\expandafter\endinput%
\fi

\begingroup
\@ifundefined{eTeXversion}{%
\PackageError{tabularht}{% 
Option 'vlines' requires eTeX%
}{% Use of eTeX is recommended for \LaTeX, see \texttt{ltnews16}.%}
%
}\endgroup

2.3 Option vlines, driver independent stuff

\begingroup
\let\@addtoreset\@gobbletwo
\newcounter{tabH@unique}%
\let\tabH@currenttab\@empty
\def\tabH@array@init{%
\ifx\@toarrayheight\@empty
% ignore vertical lines of nested tabular environments
\let\tabH@currenttab\@empty
\else
\stepcounter{tabH@unique}%
\edef\tabH@currenttab{\the\c@tabH@unique}%
\fi
}
\renewcommand*{\@arrayrule}{%
@addtopreamble{% 
\hskip -.5\arrayrulewidth
\ifx\tabH@currenttab\@empty
%
\else
\stepcounter{tabH@unique}%
\edef\tabH@currenttab{\the\c@tabH@unique}%
\fi
%
}}
\endgroup
\begin{Verbatim}
\ifpdf
\begin{group}
\PackageError{tabularht}{Your pdfTeX is too old}{\string\pdfsavepos\space is missing.}
\endgroup
\csname fi\endcsname
\endinput
\end{Verbatim}

\section*{2.4 Driver pdftex}

\verbatim
\begin{Verbatim}
\verb|\ifpdf|
\begin{group}
\PackageWarning{tabularht}{Using driver ‘pdftex’ because of pdfTeX in PDF mode}
\endgroup
% The .aux file contains three commands:
\end{Verbatim}
\AtBeginDocument{%
% The .aux files are read the first time before
% \AtBeginDocument and later at \end{document}.
% \tabH@aux@done is a marker to distinguish
% between these two readings. Only in the first
% case we need the \tabH@aux@... commands.
% \let\tabH@aux@done\@empty
\IfFileExists{%
  \immediate\write\@mainaux{%
    \@percentchar\@percentchar BeginProlog: tabularht%
  }
  % items in the aux file are executed,
  % if tabularht is loaded
  % and during the aux file read at \begin{document} only
  \immediate\write\@mainaux{%
    \detokenize{%
      % the \tabH@aux@... commands are needed only if
      % tabularht is loaded with driver pdftex.
      \@ifundefined{tabH@aux@vrule}\@secondoftwo\@firstofone
      {%
        % disable commands except for the first .aux files reading
        \@ifundefined{tabH@aux@done}\@gobble\@firstofone
        {%
          %
          \let\tabH@aux@vrule\@gobbletwo
          \let\tabH@aux@vstart\@gobblefour
          \let\tabH@aux@vstop\@gobble
        }
      }
    }
  }

  \immediate\write\@mainaux{%
    \@percentchar\@percentchar EndProlog: tabularht%
  }
}\fi
% the x positions of vrules are stored in
% \tabH@<tabcount>list with distinct values
\protected\def\tabH@aux@vrule#1#2{%
  \@ifundefined{tabH@aux@vrule@#1\#2}\@secondoftwo\@firstofone
  {%
    %
    \@ifundefined{tabH@aux@done}\@gobble\@firstofone
    {%
      %
      \let\tabH@aux@vrule\@gobbletwo
      \let\tabH@aux@vstart\@gobblefour
      \let\tabH@aux@vstop\@gobble
    }
  }
  {%
    \immediate\write\@mainaux{%
      \@percentchar\@percentchar BeginProlog: tabularht%
    }
    {%
      %
      \@ifundefined{tabH@aux@vrule@#1\#2}\@secondoftwo\@firstofone
      {%
        %
        \@ifundefined{tabH@aux@done}\@gobble\@firstofone
        {%
          %
          \let\tabH@aux@vrule\@gobbletwo
          \let\tabH@aux@vstart\@gobblefour
          \let\tabH@aux@vstop\@gobble
        }
      }
    }
  }
}\begingroup
  \def\x{#2}-%
  \let\y\@undefined-%
  \let\do\tabH@do@add-%
  \expandafter\xdef\csname tabH@#1list\endcsname{%
    \csname tabH@#1list\endcsname\@empty-%
    \ifx\y\@undefined-%
      \noexpand\do{\x}%-#2
    \fi
  }
\endgroup
\endgroup
\def\tabH@do@add#1{% 
  \ifx\y\@undefined 
    \ifnum#1<\x\space 
      \else 
        \expandafter\ifx\csname y\endcsname\relax\fi 
        \ifnum#1>\x\space 
          \noexpand\do{\x}\% 
        \fi 
      \fi 
  \fi 
  \noexpand\do{#1}\% 
}\}% 
\def\tabH@vspace@start#1{% 
  \if@filesw 
    \stepcounter{tabH@unique}\% 
    \edef\tabH@currentrow{\the\c@tabH@unique}\% 
    \pdfsavepos 
    \protected@write\@auxout{% 
      \let\tabH@lastxpos\relax 
      \let\tabH@lastypos\relax 
    }{% 
      \tabH@aux@vstart{\tabH@currenttab}{\tabH@currentrow}\% 
      {\tabH@lastxpos}{\tabH@lastypos}\% 
    }\% 
  \fi 
}\begingroup 
  \edef\a{tabH@\tabH@currenttab row\tabH@currentrow}\% 
  \expandafter\let\expandafter\x\csname\a x\endcsname \ifx\x\relax \else \expandafter\let\expandafter\y\csname\a y\endcsname \expandafter\let\expandafter\l\csname tabH@\tabH@currenttab list\endcsname \ifx\l\relax \else \def\f{#1}\% \ifx\f\@empty \let\do\tabH@do@set \else \count@=\z@ \let\do\tabH@do@filter \fi \setbox\z@=\hbox{\l}\% \wd\z@=\z@ \dp\z@=\z@ \copy\z@ \fi \fi \fi 
\endgroup}
\def\tabH@vspace@stop{% 
  \if@filesw 
    \pdfsavepos 
    \protected@write\@auxout{% 
  }\% 
}\if@filesw 
\pdfsavepos 
\protected@write\@auxout{% 

2.5 DVI drivers

\ifx\tabH@driver\empty
\PackageError{tabularht}{}{Missing DVI driver, option ‘vlines’ disabled}
\fi

\def\tabH@driver@dvips{%
516  userdict begin
517  /tabH.list 10 dict def
518  /tabH.job [] def %
519  end
520  /tabH.vrule{%
521  10 string cvs cvn dup tabH.list exch known{%
522  tabH.list exch dup [ exch tabH.list exch get %
523  currentpoint pop round exch true exch{%
524  tabH.list key [ ... x true i
525  tabH.list key [ ... false i
526  exch{
527  ... [ ... x i
528  2 copy lt{false}{%
529  2 copy eq{pop false}{exch true}ifelse%
530  }ifelse%
531  }forall %
532  pop%
533  ]put%
534  }%
535  tabH.list exch[currentpoint pop round]put%
536  }ifelse%
537 }bind def
538
539  % <tab num> <cols array> <ytop> <ybottom> <rulewidth[sp]>
540  /tabH.vspace{%
541  userdict begin %
542  10 dict dup begin %
543  exch 65536 div Resolution mul 72.27 div %
544  % dvips uses a poor man's ceil function
545  % see dopage.c before "drawrule": (int)(... + 0.9999999)
546  0.9999999 add truncate%
547  /rulewidth exch def %
548  exch/ybottom exch def %
549  exch/ytop exch def %
550  exch/cols exch def %
551  exch/tabkey exch 10 string cvs cvn def %
552  end%
553  /tabH.job exch[exch userdict/tabH.job getaload pop]def %
554  end%
555 }bind def %
556 % Now we do the work at the end of the page.
557 % Unhappily "eop-hook" cannot be used, because "eop"
558 % executes "restore" before, so that all data are lost.
559 TeXDICT begin%
560  /eop%
561  [%
562  tabH.job{%
563  begin%
564  /colarray %
565  tabH.list tabkey known{tabH.list tabkey get}{[]}ifelse %
566  def %
567  cols length 0 eq not{%
568  /colarray[%
569  cols{1 sub %
570  dup 0 lt{pop}{%
571  dup colarray length ge{pop}{%
572  colarray exch get%
\section*{3 Installation}

\subsection*{3.1 Download}

\textbf{Package.} This package is available on CTAN\textsuperscript{1}:

\begin{itemize}
  \item \texttt{CTAN:macros/latex/contrib/oberdiek/tabularht.dtx} The source file.
  \item \texttt{CTAN:macros/latex/contrib/oberdiek/tabularht.pdf} Documentation.
\end{itemize}

\textbf{Bundle.} All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

\texttt{CTAN:install/macros/latex/contrib/oberdiek.tds.zip}

\textit{TDS} refers to the standard “A Directory Structure for \TeX\ Files” (CTAN:pkg/tds). Directories with \texttt{texmf} in their name are usually organized this way.

\subsection*{3.2 Bundle installation}

\textbf{Unpacking.} Unpack the \texttt{oberdiek.tds.zip} in the TDS tree (also known as \texttt{texmf} tree) of your choice. Example (linux):

\begin{verbatim}
unzip oberdiek.tds.zip -d "~/texmf"
\end{verbatim}

\subsection*{3.3 Package installation}

\textbf{Unpacking.} The \texttt{.dtx} file is a self-extracting docstrip archive. The files are extracted by running the \texttt{.dtx} through plain \TeX:

\begin{verbatim}
tex tabularht.dtx
\end{verbatim}

\textsuperscript{1}CTAN:pkg/tabularht
TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as texmf tree):

- `tabularht.sty` → `tex/latex/oberdiek/tabularht.sty`
- `tabularht.pdf` → `doc/latex/oberdiek/tabularht.pdf`
- `tabularht-example1.tex` → `doc/latex/oberdiek/tabularht-example1.tex`
- `tabularht-example2.tex` → `doc/latex/oberdiek/tabularht-example2.tex`
- `tabularht.dtx` → `source/latex/oberdiek/tabularht.dtx`

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

3.4 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

Unpacking with LATEX. The `.dtx` chooses its action depending on the format:

**plain TEX:** Run `docstrip` and extract the files.

**LATEX:** Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{tabularht.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdflatex:

```
pdflatex tabularht.dtx
makeindex -s gind.ist tabularht.idx
pdflatex tabularht.dtx
makeindex -s gind.ist tabularht.idx
pdflatex tabularht.dtx
```

4 History

[2005/09/22 v1.0]

- First public version.
[2005/10/16 v2.0]
- Height specification allows to... or spread=..., default is to=.
- Option vlines added, drivers pdftex and dvips.
- \interrowspace, \interrowfil, and \interrowstart...\interrowstop added.

[2005/10/18 v2.1]
- Fix for package colortbl, but the colors of colortbl remain unsupported.

[2006/02/20 v2.2]
- Code is not changed.
- DTX framework.

[2006/12/22 v2.3]
- Documentation fix.
- Fix in code of option vlines.

[2007/03/21 v2.4]
- Fix: Counter tabh@unique must not be changed by \include.

[2007/04/11 v2.5]
- Line ends sanitized.

[2016/05/16 v2.6]
- Documentation updates.

[2019/12/29 v2.7]
- Use \iftex package.

5 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols

\@array 72, 73
\@addtopreamble 227 \@firstofone 308, 311
\@addtoreset 211 \@for 419, 497
\@array 58, 68, 72, 73 \@gobble 169, 250, 311, 316, 427
\@arrayrule 226, 243, 246 \@gobblefour 315
\@auxout 274, 365, 401 \@gobbletwo 211, 314
\@depth 412 \@ifpackageloaded 82, 84, 245
\@empty 51, 52, 64, \@ifundefined