

# The `tabularht` package

Heiko Oberdiek\*

2016/05/16 v2.6

## Abstract

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

## Contents

<b>1</b>	<b>Usage</b>	<b>2</b>
1.1	Option <code>vlines</code>	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
<b>2</b>	<b>Implementation</b>	<b>4</b>
2.1	Environments	4
2.2	Options	7
2.3	Option <code>vlines</code> , driver independent stuff	7
2.4	Driver <code>pdftex</code>	8
2.5	DVI drivers	11
<b>3</b>	<b>Installation</b>	<b>14</b>
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
<b>4</b>	<b>History</b>	<b>15</b>
	[2005/09/22 v1.0]	15
	[2005/10/16 v2.0]	16
	[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
<b>5</b>	<b>Index</b>	<b>16</b>

---

\*Please report any issues at <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:

- `tabularht`, `tabularht*`
- `arrayht`
- `tabularhtx` (if package `tabularx` is loaded)

The height argument allows a length specification, package `calc` is supported if used. This means, the tabular will have the specified height. You can also use the prefixes `to=` and `spread=`. `to=` is the default, `spread=` means, the natural height of the tabular box is changed by the length after `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 `tabularkv`, it provides an interface, where most parameters for the environments can be given by key-value pairs.

`\interrowspace{...}`

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

`\interrowfill`

Short for `\interrowspace{\fill}`

`\interrowstart... \interrowstop`

Marker commands, useful for option `vlines`.

## 1.1 Option `vlines`

Warning: This stuff is experimental.

Vertical lines are interrupted, if space is inserted in `\noalign`, `\interrowspace`, `\addlinespace` (`booktabs`), between double `\hlines`. This option tries to detect and add the vertical lines. The lines in a tabular with `tabularht` support (environments of this package) are numbered from left to right. The gap that is controlled by `\interrowspace` or inbetween `\interrowstart` and `\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{tabularht}{50mm}{|1|1|}
  Hello & World\\
  \interrowfill[1,3]
  Foo & Bar
\end{tabularht}
```

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  $\langle list \rangle$ .  $\langle list \rangle$  is a comma separated list of numbers,  $\langle height \rangle$  means the height specification described above with the optional prefixes **to=** or **spread=**.

```
\interrowSPACE [ $\langle list \rangle$ ] { $\langle height \rangle$ }
\interrowfill [ $\langle list \rangle$ ]
\interrowstart [ $\langle list \rangle$ ] ... \interrowstop
```

Option `vlines` is driver dependent and uses  $\varepsilon$ -TeX features.

**pdftex:** pdfTeX in PDF mode. Here the positions of the lines are written with the help of the `\pdfsavepos` feature into the `.aux` file(s). Therefore you need two LaTeX runs to get the lines.

**dvips:** Here, PostScript's `currentpoint` is 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.

**Other drivers:**

**PostScript drivers:** probably possible, an end of page hook would be nice.

**VT<sub>E</sub>X:** with GeX (PostScript interpreter) probably possible.

**dvipdfm:** no idea. The big problem is, how to get the current position?

## 1.2 Limitations

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

## 1.3 Compatibility

- `array`, `delarray`, `tabularx` are supported.
- There can be problems with packages that redefine `\@array` (or `\@@array`, `\@tabarray`) and `\@arrayrule` (for option `vlines`).
- `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 \documentclass{article}
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}%
10    \multicolumn{2}{@{}c@{}}{bounding box}\\%
11    \noalign{\vfill}%
12 }
```

```

12   lower left corner & lower right corner\\%
13   \end{tabularht*}%
14 }
15 \end{document}
16 \end{example1}

```

### 1.4.2 Example 2

```

17 \begin{example2}
18 \documentclass{article}
19 \usepackage{booktabs}
20 \usepackage[dvips,vlines]{tabularht}
21
22 \begin{document}
23
24 \begin{tabularht}{spread=0pt}{|l|l|}
25   \hline
26   First&Line\\%
27   \hline
28 \interrowstart
29   \addlinespace[10mm]%
30 \interrowstop
31   \hline
32   Second&Line\\%
33 \interrowstart
34   \hline
35   \hline
36 \interrowstop
37   Third&Line\\%
38   \hline
39 \interrowspace{10mm}
40   \hline
41   Fourth&Line\\%
42   \hline
43 \end{tabularht}
44
45 \end{document}
46 \end{example2}

```

## 2 Implementation

```

47 \begin{package}

```

Package identification.

```

48 \NeedsTeXFormat{LaTeX2e}
49 \ProvidesPackage{tabularht}%
50 [2016/05/16 v2.6 Tabular with height specified (HO)]

```

### 2.1 Environments

```

51 \let\toarrayheight\@empty
52 \let\tabH@array@init\@empty
53
54 \toks@={%
55   \begin{group
56     \long\def\x#1\vcenter\fi\fi\bgroup#2\@sharp#3#4\@nil{%
57       \end{group
58       \gdef\@array[##1]##2{%
59         \tabH@array@init
60         #1%
61         \vcenter\fi\fi

```

```

62      \@toarrayheight
63      \bgroup
64      \let\@toarrayheight\@empty
65      #2\@sharp###3#4%
66    }%
67  }%
68  \expandafter\x\@array[#1]{#2}\@nil % hash-ok
69 }
70 \edef\tabH@patch@array{\the\toks@}
71 \def\tabH@patch@@array{%
72   \ifx\@array\@array
73     \def\reserved@a{\let\@array\@array}%
74   \else
75     \let\reserved@a\relax
76   \fi
77   \tabH@patch@array
78   \reserved@a
79 }
80 \tabH@patch@@array
81
82 \@ifpackageloaded{array}{\{%
83   \AtBeginDocument{%
84     \@ifpackageloaded{array}{\%
85       \tabH@patch@@array
86     }\}%
87   }%
88 }
89
90 \def\tabH@setheight#1{%
91   \tabH@@setheight#1==\@nil
92 }
93 \def\tabH@@setheight#1=#2=#3\@nil{%
94   \ifx\#2#3\%
95     \setlength{\dimen@}{#1}%
96     \edef\@toarrayheight{\the\dimen@}%
97   \else
98     \edef\tabH@temp{\zap@space#1 \@empty}%
99     \ifx\tabH@temp\tabH@to
100    \else
101      \ifx\tabH@temp\tabH@spread
102      \else
103        \PackageError{tabularht}{%
104          Unknown height specifier %
105          '\expandafter\strip@prefix\meaning\tabH@temp'%
106        }{%
107          The height dimension for tabular height can be prefixed%
108          \MessageBreak
109          with 'to=' or 'spread=', default is 'to='.%
110        }%
111        \let\tabH@temp\tabH@to
112      \fi
113    \fi
114    \setlength{\dimen@}{#2}%
115    \edef\@toarrayheight{\tabH@temp\the\dimen@}%
116  \fi
117 }
118 \def\tabH@to{to}
119 \def\tabH@spread{spread}

```

First argument is the height of the table, then the original arguments for tabular follow.

```

120 \newenvironment{tabularht}[1]{%
121   \tabH@setheight{#1}%
122   \tabular
123 }{%
124   \endtabular
125 }
126
127 \newenvironment{tabularht*}[1]{%
128   \tabH@setheight{#1}%
129   \@nameuse{tabular*}%
130 }{%
131   \@nameuse{endtabular*}%
132 }
133
134 \newenvironment{tabularhtx}[1]{%
135   \tabH@setheight{#1}%
136   \tabularx
137 }{%
138   \endtabularx
139 }
140
141 \newenvironment{arrayht}[1]{%
142   \tabH@setheight{#1}%
143   \array
144 }{%
145   \endarray
146 }
147
148 \def\interrowSPACE{%
149   \noalign\bgroup
150     \tabH@interrowSPACE
151 }
152 \newcommand*\tabH@interrowSPACE}[2] [] {%
153   \tabH@vspace{#1}{#2}%
154   \egroup
155 }
156 \def\interrowfill{%
157   \noalign\bgroup
158     \tabH@interrowfill
159 }
160 \newcommand*\tabH@interrowfill}[1] [] {%
161   \tabH@vspace{#1}{\fill}%
162   \egroup
163 }
164 \def\tabH@vspace#1#2{%
165   \tabH@vspace@start{#1}%
166   \vspace{#2}%
167   \tabH@vspace@stop
168 }
169 \let\tabH@vspace@start\@gobble
170 \let\tabH@vspace@stop\@empty
171
172 \newcommand*\interrowstart{%
173   \noalign\bgroup
174     \tabH@interrowstart
175 }

```

```

176 \newcommand*{\tabH@interrowstart}[1] [] {%
177     \tabH@vspace@start{#1}%
178     \egroup
179 }
180 \newcommand*{\interrowstop}{%
181     \noalign{\tabH@vspace@stop}%
182 }

```

## 2.2 Options

```

183 \providecommand*{\tabH@driver}{}
184
185 \DeclareOption{vlines}{%
186     \let\tabH@temp\relax
187 }
188 \DeclareOption{pdftex}{}
189 \DeclareOption{dvips}{%
190     \def\tabH@driver{dvips}%
191 }
192 \ProcessOptions*\relax
193
194 \ifx\tabH@temp\relax
195 \else
196     \expandafter\endinput
197 \fi
198
199 \begingroup
200     \@ifundefined{eTeXversion}{%
201         \PackageError{tabularht}{%
202             Option ‘vlines’ requires eTeX%
203         }{%
204             Use of eTeX is recommended for LaTeX, see ltnews16.%
205         }%
206     \endgroup
207     \endinput
208 }{}%
209 \endgroup

```

## 2.3 Option vlines, driver independent stuff

```

210 \begingroup
211     \let\@addtoreset\@gobbletwo
212     \newcounter{tabH@unique}%
213 \endgroup
214 \let\tabH@currenttab\@empty
215
216 \def\tabH@array@init{%
217     \ifx\@toarrayheight\@empty
218         % ignore vertical lines of nested tabular environments
219         \let\tabH@currenttab\@empty
220     \else
221         \stepcounter{tabH@unique}%
222         \edef\tabH@currenttab{\the\c@tabH@unique}%
223     \fi
224 }
225
226 \renewcommand*{\@arrayrule}{%
227     \@addtopreamble{%
228         \hskip -.5\arrayrulewidth
229         \ifx\tabH@currenttab\@empty

```

```

230 \else
231 \tabH@vrule{\tabH@currenttab}%
232 \fi
233 \begingroup
234 \expandafter\ifx\csname CT@arc@\endcsname\relax
235 \else
236 \expandafter\CT@arc@
237 \fi
238 \vline
239 \endgroup
240 \hskip -.5\arrayrulewidth
241 }%
242 }
243 \let\tabH@arrayrule\@arrayrule
244 \AtBeginDocument{%
245 \ifpackageloaded{colortbl}{%
246 \let\@arrayrule\tabH@arrayrule
247 }{}%
248 }
249
250 \let\tabH@vrule\@gobble

```

## 2.4 Driver pdftex

```

251 \RequirePackage{ifpdf}
252 \ifpdf
253 \begingroup
254 \ifundefined{pdfsavepos}{%
255 \PackageError{tabularht}{%
256 Your pdfTeX is too old%
257 }{%
258 \string\pdfsavepos\space is missing.%
259 }%
260 \endgroup
261 \csname fi\endcsname
262 \endinput
263 }{}%
264
265 \let\on@line\@empty
266 \PackageInfo{tabularht}{%
267 Using driver 'pdftex' because of pdfTeX in PDF mode%
268 }%
269 \endgroup
270
271 \protected\def\tabH@vrule#1{%
272 \if@files@
273 \pdfsavepos
274 \protected@write\@auxout{%
275 \let\tabH@lastxpos\relax
276 }{%
277 \tabH@aux@vrule{#1}{\tabH@lastxpos}%
278 }%
279 \fi
280 }%
281
282 \def\tabH@lastxpos{\the\pdflastxpos}%
283 \def\tabH@lastypos{\the\pdflastypos}%
284
285 % The .aux file contains three commands:

```



```

286 % \tabH@aux@vrule{tabular id}{x position}
287 % \tabH@aux@vstart{tabular id}{row id}{x position}{y position}
288 % \tabH@aux@vstop{y position}
289 %
290 \AtBeginDocument{%
291   % The .aux files are read the first time before
292   % \AtBeginDocument and later at \end{document}.
293   % \tabH@aux@done is a marker to distinguish
294   % between these two readings. Only in the first
295   % case we need the \tabH@aux@... commands.
296   \let\tabH@aux@done\@empty
297   \if@filesw
298     \immediate\write\@mainaux{%
299       \@percentchar\@percentchar BeginProlog: tabularht%
300     }%
301     % items in the aux file are executed,
302     % if tabularht is loaded
303     % and during the aux file read at \begin{document} only
304     \immediate\write\@mainaux{%
305       \detokenize{%
306         % the \tabH@aux@... commands are needed only if
307         % tabularht is loaded with driver pdftex.
308         \@ifundefined{tabH@aux@vrule}\@secondoftwo\@firstofone
309       }%
310       % disable commands except for the first .aux files reading
311       \@ifundefined{tabH@aux@done}\@gobble\@firstofone
312     }%
313     {%
314       \let\tabH@aux@vrule\@gobbletwo
315       \let\tabH@aux@vstart\@gobblefour
316       \let\tabH@aux@vstop\@gobble
317     }%
318   }%
319 }%
320 \immediate\write\@mainaux{%
321   \@percentchar\@percentchar EndProlog: tabularht%
322 }%
323 \fi
324 }%
325
326 % the x positions of vrules are stored in
327 % \tabH@<tabcount>list with distinct values
328 \protected\def\tabH@aux@vrule#1#2{%
329   \@ifundefined{tabH@#1list}{%
330     \expandafter\xdef\csname tabH@#1list\endcsname{%
331       \noexpand\do{#2}%
332     }%
333   }{%
334     \begingroup
335     \def\x{#2}%
336     \let\y\@undefined
337     \let\do\tabH@do@add
338     \expandafter\xdef\csname tabH@#1list\endcsname{%
339       \csname tabH@#1list\endcsname\@empty
340       \ifx\y\@undefined
341         \noexpand\do{\x}%
342       \fi
343     }%

```

```

344     \endgroup
345 }%
346 }%
347 \def\tabH@do@add#1{%
348   \ifx\y\@undefined
349     \ifnum#1<\x\space
350     \else
351       \expandafter\ifx\csname y\endcsname\relax\fi
352       \ifnum#1>\x\space
353         \noexpand\do{\x}%
354       \fi
355     \fi
356   \fi
357   \noexpand\do{#1}%
358 }%
359
360 \def\tabH@vspace@start#1{%
361   \if@files
362     \stepcounter{tabH@unique}%
363     \edef\tabH@currentrow{\the\c@tabH@unique}%
364     \pdfsavepos
365     \protected@write\@auxout{%
366       \let\tabH@lastxpos\relax
367       \let\tabH@lastypos\relax
368     }{%
369       \tabH@aux@vstart{\tabH@currenttab}{\tabH@currentrow}%
370       {\tabH@lastxpos}{\tabH@lastypos}%
371     }%
372   \fi
373   \begingroup
374     \edef\@{tabH@\tabH@currenttab row\tabH@currentrow}%
375     \expandafter\let\expandafter\x\csname\@ x\endcsname
376     \ifx\x\relax
377     \else
378       \expandafter\let\expandafter\y\csname\@ y\endcsname
379       \expandafter\let\expandafter\l
380       \csname tabH@\tabH@currenttab list\endcsname
381       \ifx\l\relax
382       \else
383         \def\f{#1}%
384         \ifx\f\@empty
385           \let\do\tabH@do@set
386         \else
387           \count@=\z@
388           \let\do\tabH@do@filter
389         \fi
390         \setbox\z@=\hbox{\l}%
391         \wd\z@=\z@
392         \dp\z@=\z@
393         \copy\z@
394       \fi
395     \fi
396   \endgroup
397 }%
398 \def\tabH@vspace@stop{%
399   \if@files
400     \pdfsavepos
401     \protected@write\@auxout{%

```

```

402     \let\tabH@lastypos\relax
403   }{%
404     \tabH@aux@vstop{\tabH@lastypos}%
405   }%
406   \fi
407 }%
408 \def\tabH@do@set#1{%
409   \hbox to \z@{%
410     \hskip \dimexpr #1sp - \x sp\relax
411     \vrule \@width\arrayrulewidth
412           \@depth\dimexpr \y sp\relax
413     \hss
414   }%
415 }%
416 \def\tabH@do@filter{%
417   \@tempswafalse
418   \advance\count@ \@ne
419   \@for\@e:=\f\do{%
420     \ifnum\@e=\count@
421       \@tempswatrue
422     \fi
423   }%
424   \if@tempswa
425     \expandafter\tabH@do@set
426   \else
427     \expandafter\@gobble
428   \fi
429 }%
430
431 \protected\def\tabH@aux@vstart#1#2#3#4{%
432   \def\tabH@current@vstart{{#1}{#2}{#3}{#4}}%
433 }%
434 \protected\def\tabH@aux@vstop{%
435   \expandafter\tabH@aux@v\tabH@current@vstart
436 }%
437 \def\tabH@aux@v#1#2#3#4#5{%
438   \expandafter\gdef\csname tabH@#1row#2x\endcsname{#3}%
439   \expandafter\xdef\csname tabH@#1row#2y\endcsname{%
440     \the\numexpr #4 - #5\relax
441   }%
442 }%
443
444 \csname fi\endcsname
445 \endinput
446
447 \fi

```

## 2.5 DVI drivers

```

448 \ifx\tabH@driver\empty
449   \PackageError{tabularht}{%
450     Missing DVI driver, option ‘vlines’ disabled%
451   }{%
452     Supported DVI drivers: dvips.%
453   }%
454   \expandafter\endinput
455 \fi
456
457 \def\tabH@driver@dvips{%

```

```

458 \def\tabH@literalps##1{\special{ps:SDict begin ##1 end}}%
459 \def\tabH@headerps##1{\special{! ##1}}%
460 }
461
462 \@onelevel@sanitize\tabH@driver
463 \@ifundefined{tabH@driver}\tabH@driver}{%
464 \PackageError{tabularht}{%
465   Unsupported driver '\tabH@driver'%
466 }{%
467   Supported DVI drivers: dvips.%
468 }%
469 \endinput
470 }{}
471
472 \begingroup
473 \let\on@line\@empty
474 \PackageInfo{tabularht}{%
475   Using driver '\tabH@driver'%
476 }%
477 \endgroup
478 \csname tabH@driver@\tabH@driver\endcsname
479
480 \protected\def\tabH@vrule#1#2\vrule#3\arrayrulewidth{%
481   #2% \fi or empty
482   % hack to get rid of maxdrift rounding of dvips,
483   % thus simulate a large motion
484   \kern1in\relax
485   \tabH@literalps{%
486     #1 tabH.vrule %
487     Resolution neg 0 translate%
488   }%
489   \vrule#3\arrayrulewidth
490   \tabH@literalps{Resolution 0 translate}%
491   \kern-1in\relax
492 }
493
494 \def\tabH@vspace@start#1{%
495   \begingroup
496   \let\y\@empty
497   \@for\x:=#1\do{%
498     \ifx\y\@empty
499       \edef\y{\x}%
500     \else
501       \edef\y{\y\space\x}%
502     \fi
503   }%
504   \tabH@literalps{\tabH@currenttab[\y]currentpoint exch pop}%
505   \endgroup
506 }
507 \def\tabH@vspace@stop{%
508   \tabH@literalps{%
509     currentpoint exch pop %
510     \number\dimexpr\arrayrulewidth\relax\space
511     tabH.vspace%
512   }%
513 }
514
515 \tabH@headerps{%

```

```

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       }{false}ifelse%
532     }forall %
533     pop%
534     ]put%
535   }{%
536     tabH.list exch[currentpoint pop round]put%
537   }ifelse%
538 }bind def%
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 get aload 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   {%
563     tabH.job{%
564       begin%
565         /colarray %
566         tabH.list tabkey known{tabH.list tabkey get}{}ifelse %
567         def %
568         cols length 0 eq not{%
569           /colarray[%
570             cols{1 sub %
571               dup 0 lt{pop}{%
572                 dup colarray length ge{pop}{%
573                   colarray exch get%

```

```

574             }ifelse%
575         }ifelse%
576     }forall%
577     ]def%
578     }if %
579     colarray{%
580         % (rulewidth) == rulewidth == % debug
581         Resolution sub %
582         ytop rulewidth ytop ybottom sub v%
583     }forall %
584     end%
585     }forall%
586     % tabH.list{== ==}forall % debug
587     }bind aload pop %
588     TeXDict /eop get aload pop%
589     ]cvx def %
590 end%
591 }
592 </package>

```

## 3 Installation

### 3.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[CTAN:macros/latex/contrib/oberdiek/tabularht.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/tabularht.pdf](#) Documentation.

**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.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

*TDS* refers to the standard “A Directory Structure for T<sub>E</sub>X Files” ([CTAN:pkg/tds](#)). Directories with `texmf` in their name are usually organized this way.

### 3.2 Bundle installation

**Unpacking.** Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

### 3.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain T<sub>E</sub>X:

```
tex tabularht.dtx
```

---

<sup>1</sup>[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  $\text{\TeX}$  distribution ( $\text{\TeX}$ Live, MiK $\text{\TeX}$ , ...) relies on file name databases, you must refresh these. For example,  $\text{\TeX}$ Live users run `texhash` or `mktextlsr`.

### 3.5 Some details for the interested

**Unpacking with  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain  $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), 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 `pdf $\text{\LaTeX}$` :

```
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`.
- `\interrow space`, `\interrow fil`, and `\interrow start... \interrow stop` 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.

## 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	
<code>\@@array</code> . . . . .	<i>72, 73</i>
<code>\@addtopreamble</code> . . . . .	<i>227</i>
<code>\@addtoreset</code> . . . . .	<i>211</i>
<code>\@array</code> . . . . .	<i>58, 68, 72, 73</i>
<code>\@arrayrule</code> . . . . .	<i>226, 243, 246</i>
<code>\@auxout</code> . . . . .	<i>274, 365, 401</i>
<code>\@depth</code> . . . . .	<i>412</i>
<code>\@empty</code> . . . . .	<i>51, 52, 64,</i> <i>98, 170, 214, 217, 219, 229, 265,</i> <i>296, 339, 384, 448, 473, 496, 498</i>
<code>\@firstofone</code> . . . . .	<i>308, 311</i>
<code>\@for</code> . . . . .	<i>419, 497</i>
<code>\@gobble</code> . . . . .	<i>169, 250, 311, 316, 427</i>
<code>\@gobblefour</code> . . . . .	<i>315</i>
<code>\@gobbletwo</code> . . . . .	<i>211, 314</i>
<code>\@ifpackageloaded</code> . . . . .	<i>82, 84, 245</i>
<code>\@ifundefined</code> . . . . .	<i>200, 254, 308, 311, 329, 463</i>
<code>\@mainaux</code> . . . . .	<i>298, 304, 320</i>
<code>\@nameuse</code> . . . . .	<i>129, 131</i>
<code>\@ne</code> . . . . .	<i>418</i>
<code>\@nil</code> . . . . .	<i>56, 68, 91, 93</i>
<code>\@onelevel@sanitize</code> . . . . .	<i>462</i>
<code>\@percentchar</code> . . . . .	<i>299, 321</i>
<code>\@secondoftwo</code> . . . . .	<i>308</i>





<code>\space</code> .....	258, 349, 352, 501, 510	<code>\tabH@temp</code> .....	98,
<code>\special</code> .....	458, 459		99, 101, 105, 111, 115, 186, 194
<code>\stepcounter</code> .....	221, 362	<code>\tabH@to</code> .....	99, 111, 118
<code>\strip@prefix</code> .....	105	<code>\tabH@vrule</code> .....	231, 250, 271, 480
<b>T</b>		<code>\tabH@vspace</code> .....	153, 161, 164
<code>\tabH@</code> .....	327	<code>\tabH@vspace@start</code> .....	
<code>\tabH@@setheight</code> .....	91, 93		165, 169, 177, 360, 494
<code>\tabH@array@init</code> .....	52, 59, 216	<code>\tabH@vspace@stop</code> .....	
<code>\tabH@arrayrule</code> .....	243, 246		167, 170, 181, 398, 507
<code>\tabH@aux@</code> .....	295, 306	<code>\tabular</code> .....	122
<code>\tabH@aux@done</code> .....	293, 296	<code>\tabularx</code> .....	136
<code>\tabH@aux@v</code> .....	435, 437	<code>\the</code> .	70, 96, 115, 222, 282, 283, 363, 440
<code>\tabH@aux@vrule</code> ...	277, 286, 314, 328	<code>\toks@</code> .....	54, 70
<code>\tabH@aux@vstart</code> ..	287, 315, 369, 431	<b>U</b>	
<code>\tabH@aux@vstop</code> ...	288, 316, 404, 434	<code>\usepackage</code> .....	3, 19, 20
<code>\tabH@current@vstart</code> .....	432, 435	<b>V</b>	
<code>\tabH@currentrow</code> .....	363, 369, 374	<code>\vcenter</code> .....	56, 61
<code>\tabH@currenttab</code> .....	214, 219,	<code>\vfill</code> .....	9, 11
	222, 229, 231, 369, 374, 380, 504	<code>\vline</code> .....	238
<code>\tabH@do@add</code> .....	337, 347	<code>\vrule</code> .....	411, 480, 489
<code>\tabH@do@filter</code> .....	388, 416	<code>\vspace</code> .....	166
<code>\tabH@do@set</code> .....	385, 408, 425	<b>W</b>	
<code>\tabH@driver</code> .....	183,	<code>\wd</code> .....	391
	190, 448, 462, 463, 465, 475, 478	<code>\write</code> .....	298, 304, 320
<code>\tabH@driver@dvi</code> .....	457	<b>X</b>	
<code>\tabH@headerps</code> .....	459, 515	<code>\x</code> .....	56, 68, 335, 341, 349, 352,
<code>\tabH@interrowfill</code> .....	158, 160		353, 375, 376, 410, 497, 499, 501
<code>\tabH@interrow</code> .....	150, 152	<b>Y</b>	
<code>\tabH@interrowstart</code> .....	174, 176	<code>\y</code> .....	336, 340, 348,
<code>\tabH@lastxpos</code> .....	275, 277, 282, 366, 370		378, 412, 496, 498, 499, 501, 504
<code>\tabH@lastypos</code> .....	283, 367, 370, 402, 404	<b>Z</b>	
<code>\tabH@literalps</code> .....	458, 485, 490, 504, 508	<code>\z@</code> .....	387, 390, 391, 392, 393, 409
<code>\tabH@patch@@array</code> .....	71, 80, 85	<code>\zap@space</code> .....	98
<code>\tabH@patch@array</code> .....	70, 77		
<code>\tabH@setheight</code> .....	90, 121, 128, 135, 142		
<code>\tabH@spread</code> .....	101, 119		