

# The `tabularht` package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2007/04/11 v2.5

## Abstract

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

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Usage</b>  | <b>1</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   | 3         |
| <b>2</b> | <b>Implementation</b>                                 | <b>4</b>  |
| 2.1      | Environments  | 4         |
| 2.2      | Options   | 6         |
| 2.3      | Option <code>vlines</code> , driver independent stuff | 7         |
| 2.4      | Driver <code>pdftex</code>                            | 7         |
| 2.5      | DVI drivers   | 11        |
| <b>3</b> | <b>Installation</b>                                   | <b>13</b> |
| 3.1      | Download  | 13        |
| 3.2      | Bundle installation                                   | 13        |
| 3.3      | Package installation                                  | 14        |
| 3.4      | Refresh file name databases                           | 14        |
| 3.5      | Some details for the interested                       | 14        |
| <b>4</b> | <b>History</b>  | <b>15</b> |
|          | [2005/09/22 v1.0]                                     | 15        |
|          | [2005/10/16 v2.0]                                     | 15        |
|          | [2005/10/18 v2.1]                                     | 15        |
|          | [2006/02/20 v2.2]                                     | 15        |
|          | [2006/12/22 v2.3]                                     | 15        |
|          | [2007/03/21 v2.4]                                     | 15        |
|          | [2007/04/11 v2.5]                                     | 15        |
| <b>5</b> | <b>Index</b>  | <b>15</b> |

## 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 *<list>*. *<list>* is a comma separated list of numbers, *<height>* means the height specification described above with the optional prefixes `to=` or `spread=`.

```

\interrow space [⟨list⟩] {⟨height⟩}
\interrow fill [⟨list⟩]
\interrow start [⟨list⟩] ... \interrow stop

```

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>EX</sub>:** 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 ⟨*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}%
10    \multicolumn{2}{@{}c@{}}{bounding box}\\%
11    \noalign{\vfill}%
12    lower left corner & lower right corner\\%
13  \end{tabularht*}%
14 }
15 \end{document}
16 ⟨/example1⟩

```

### 1.4.2 Example 2

```

17 ⟨*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 (*package)

```

Package identification.

```

48 \NeedsTeXFormat{LaTeX2e}
49 \ProvidesPackage{tabularht}%
50   [2007/04/11 v2.5 Tabular with height specification (H0)]

```

### 2.1 Environments

```

51 \let\toarrayheight\@empty
52 \let\tabH@array@init\@empty
53
54 \toks@={%
55   \begingroup
56     \long\def\x#1\center\fi\fi\bgroup#2\@sharp#3#4\@nil{%
57       \endgroup
58       \gdef\@array[##1]##2{%
59         \tabH@array@init
60         #1%
61         \center\fi\fi
62         \@arrayheight
63         \bgroup
64         \let\toarrayheight\@empty
65         #2\@sharp###3#4%
66       }%
67     }%
68   \expandafter\x\@array[#1]{#2}\@nil
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   \tracingmacros=1
92   \tabH@@setheight#1==\@nil
93 }
94 \def\tabH@@setheight#1=#2=#3\@nil{%
95   \ifx\#2#3\%
96     \setlength{\dimen@}{#1}%
97     \edef\@toarrayheight{to\the\dimen@}%
98   \else
99     \edef\tabH@temp{\zap@space#1 \@empty}%
100    \ifx\tabH@temp\tabH@to
101    \else
102      \ifx\tabH@temp\tabH@spread
103      \else
104        \PackageError{tabularht}{-}{%
105          Unknown height specifier %
106          '\expandafter\strip@prefix\meaning\tabH@temp'%
107        }-{%
108          The height dimension for tabular height can be prefixed%
109          \MessageBreak
110          with 'to=' or 'spread=', default is 'to='.%
111        }%
112        \let\tabH@temp\tabH@to
113      \fi
114    \fi
115    \setlength{\dimen@}{#2}%
116    \edef\@toarrayheight{\tabH@temp\the\dimen@}%
117  \fi
118 }
119 \def\tabH@to{to}
120 \def\tabH@spread{spread}

First argument is the height of the table, then the original arguments for tabular
follow.
121 \newenvironment{tabularht}[1]{-}{%
122   \tabH@setheight{#1}%
123   \tabular
124 }-{%
125   \endtabular
126 }
127
128 \newenvironment{tabularht*}[1]{-}{%
129   \tabH@setheight{#1}%
130   \@nameuse{tabular*}%
131 }-{%
132   \@nameuse{endtabular*}%
133 }
134
135 \newenvironment{tabularhtx}[1]{-}{%
136   \tabH@setheight{#1}%
137   \tabularx
138 }-{%

```

```

139 \endtabularx
140 }
141
142 \newenvironment{arrayht}[1]{%
143   \tabH@setheight{#1}%
144   \array
145 }{%
146   \endarray
147 }
148
149 \def\interrowSPACE{%
150   \noalign\bgroup
151     \tabH@interrowSPACE
152 }
153 \newcommand*{\tabH@interrowSPACE}[2][]{%
154   \tabH@vSPACE{#1}{#2}%
155   \egroup
156 }
157 \def\interrowFILL{%
158   \noalign\bgroup
159     \tabH@interrowFILL
160 }
161 \newcommand*{\tabH@interrowFILL}[1][]{%
162   \tabH@vSPACE{#1}{\fill}%
163   \egroup
164 }
165 \def\tabH@vSPACE#1#2{%
166   \tabH@vSPACE@start{#1}%
167   \vSPACE{#2}%
168   \tabH@vSPACE@stop
169 }
170 \let\tabH@vSPACE@start\@gobble
171 \let\tabH@vSPACE@stop\@empty
172
173 \newcommand*{\interrowstart}{%
174   \noalign\bgroup
175     \tabH@interrowstart
176 }
177 \newcommand*{\tabH@interrowstart}[1][]{%
178   \tabH@vSPACE@start{#1}%
179   \egroup
180 }
181 \newcommand*{\interrowstop}{%
182   \noalign{\tabH@vSPACE@stop}%
183 }

```

## 2.2 Options

```

184 \providecommand*{\tabH@driver}{}
185
186 \DeclareOption{vlines}{%
187   \let\tabH@temp\relax
188 }
189 \DeclareOption{pdftex}{}
190 \DeclareOption{dvips}{%
191   \def\tabH@driver{dvips}%
192 }
193 \ProcessOptions*\relax
194
195 \ifx\tabH@temp\relax
196 \else
197   \expandafter\endinput
198 \fi

```

```

199
200 \begingroup
201   \@ifundefined{eTeXversion}{%
202     \PackageError{tabularht}{%
203       Option 'vlines' requires eTeX%
204     }{%
205       Use of eTeX is recommended for LaTeX, see ltnews16.%
206     }%
207   \endgroup
208   \endinput
209 }{}%
210 \endgroup

```

## 2.3 Option vlines, driver independent stuff

```

211 \begingroup
212   \let\@addtoreset\@gobbletwo
213   \newcounter{tabH@unique}%
214 \endgroup
215 \let\tabH@currenttab\@empty
216
217 \def\tabH@array@init{%
218   \ifx\@toarrayheight\@empty
219     % ignore vertical lines of nested tabular environments
220     \let\tabH@currenttab\@empty
221   \else
222     \stepcounter{tabH@unique}%
223     \edef\tabH@currenttab{\the\c@tabH@unique}%
224   \fi
225 }
226
227 \renewcommand*{\@arrayrule}{%
228   \@addtopreamble{%
229     \hskip -.5\arrayrulewidth
230     \ifx\tabH@currenttab\@empty
231       \else
232         \tabH@vrule{\tabH@currenttab}%
233       \fi
234     \begingroup
235       \expandafter\ifx\csname CT@arc@\endcsname\relax
236       \else
237         \expandafter\CT@arc@
238       \fi
239     \vline
240     \endgroup
241     \hskip -.5\arrayrulewidth
242   }%
243 }
244 \let\tabH@arrayrule\@arrayrule
245 \AtBeginDocument{%
246   \@ifpackageloaded{colortbl}{%
247     \let\@arrayrule\tabH@arrayrule
248   }{}%
249 }
250
251 \let\tabH@vrule\@gobble

```

## 2.4 Driver pdftex

```

252 \RequirePackage{ifpdf}
253 \ifpdf
254   \begingroup
255     \@ifundefined{pdfsavepos}{%
256       \PackageError{tabularht}{%

```

```

257     Your pdfTeX is too old%
258   }{%
259     \string\pdfsavepos\space is missing.%
260   }%
261   \endgroup
262   \csname fi\endcsname
263   \endinput
264 }{%
265
266   \let\on@line\@empty
267   \PackageInfo{tabularht}{%
268     Using driver 'pdftex' because of pdfTeX in PDF mode%
269   }%
270 \endgroup
271
272 \protected\def\tabH@vrule#1{%
273   \if@filesw
274     \pdfsavepos
275     \protected@write\@auxout{%
276       \let\tabH@lastxpos\relax
277     }{%
278       \tabH@aux@vrule{#1}{\tabH@lastxpos}%
279     }%
280   \fi
281 }%
282
283 \def\tabH@lastxpos{\the\pdflastxpos}%
284 \def\tabH@lastypos{\the\pdflastypos}%
285
286 % The .aux file contains three commands:
287 % \tabH@aux@vrule{tabular id}{x position}
288 % \tabH@aux@vstart{tabular id}{row id}{x position}{y position}
289 % \tabH@aux@vstop{y position}
290 %
291 \AtBeginDocument{%
292   % The .aux files are read the first time before
293   % \AtBeginDocument and later at \end{document}.
294   % \tabH@aux@done is a marker to distinguish
295   % between these two readings. Only in the first
296   % case we need the \tabH@aux@... commands.
297   \let\tabH@aux@done\@empty
298   \if@filesw
299     \immediate\write\@mainaux{%
300       \@percentchar\@percentchar BeginProlog: tabularht%
301     }%
302     % items in the aux file are executed,
303     % if tabularht is loaded
304     % and during the aux file read at \begin{document} only
305     \immediate\write\@mainaux{%
306       \detokenize{%
307         % the \tabH@aux@... commands are needed only if
308         % tabularht is loaded with driver pdftex.
309         \@ifundefined{tabH@aux@vrule}\@secondoftwo\@firstofone
310       }%
311       % disable commands except for the first .aux files reading
312       \@ifundefined{tabH@aux@done}\@gobble\@firstofone
313     }%
314     {%
315       \let\tabH@aux@vrule\@gobbletwo
316       \let\tabH@aux@vstart\@gobblefour
317       \let\tabH@aux@vstop\@gobble
318     }%

```



```

319     }%
320     }%
321     \immediate\write\@mainaux{%
322     \@percentchar\@percentchar EndProlog: tabularht%
323     }%
324     \fi
325 }%
326
327 % the x positions of vrules are stored in
328 % \tabH@<tabcount>list with distinct values
329 \protected\def\tabH@aux@vrule#1#2{%
330     \ifundefined\tabH@#1list{%
331         \expandafter\xdef\csname tabH@#1list\endcsname{%
332             \noexpand\do{#2}%
333         }%
334     }{%
335         \beginngroup
336         \def\x{#2}%
337         \let\y\@undefined
338         \let\do\tabH@do@add
339         \expandafter\xdef\csname tabH@#1list\endcsname{%
340             \csname tabH@#1list\endcsname\@empty
341             \ifx\y\@undefined
342                 \noexpand\do{\x}%
343             \fi
344         }%
345         \endgroup
346     }%
347 }%
348 \def\tabH@do@add#1{%
349     \ifx\y\@undefined
350         \ifnum#1<\x\space
351         \else
352             \expandafter\ifx\csname y\endcsname\relax\fi
353             \ifnum#1>\x\space
354                 \noexpand\do{\x}%
355             \fi
356         \fi
357     \fi
358     \noexpand\do{#1}%
359 }%
360
361 \def\tabH@vspace@start#1{%
362     \if@filesw
363         \stepcounter\tabH@unique}%
364     \edef\tabH@currentrow{\the\c@tabH@unique}%
365     \pdfsavepos
366     \protected@write\@auxout{%
367         \let\tabH@lastxpos\relax
368         \let\tabH@lastypos\relax
369     }{%
370         \tabH@aux@vstart{\tabH@currenttab}{\tabH@currentrow}%
371         {\tabH@lastxpos}{\tabH@lastypos}%
372     }%
373     \fi
374     \beginngroup
375     \edef\@{tabH@\tabH@currenttab row\tabH@currentrow}%
376     \expandafter\let\expandafter\x\csname\@ x\endcsname
377     \ifx\x\relax
378     \else
379         \expandafter\let\expandafter\y\csname\@ y\endcsname
380         \expandafter\let\expandafter\l

```

```

381         \csname tabH@\tabH@currenttab list\endcsname
382     \ifx\l\relax
383     \else
384         \def\{#1}%
385         \ifx\f\@empty
386             \let\do\tabH@do@set
387         \else
388             \count@=\z@
389             \let\do\tabH@do@filter
390         \fi
391         \setbox\z@=\hbox{\l}%
392         \wd\z@=\z@
393         \dp\z@=\z@
394         \copy\z@
395     \fi
396 \fi
397 \endgroup
398 }%
399 \def\tabH@vspace@stop{%
400     \if@filesw
401         \pdfsavepos
402         \protected@write\@auxout{%
403             \let\tabH@lastypos\relax
404             }{%
405                 \tabH@aux@vstop{\tabH@lastypos}%
406             }%
407     \fi
408 }%
409 \def\tabH@do@set#1{%
410     \hbox to \z@{%
411         \hskip \dimexpr #1sp - \x sp\relax
412         \vrule \@width\arrayrulewidth
413             \@depth\dimexpr \y sp\relax
414     \hss
415     }%
416 }%
417 \def\tabH@do@filter{%
418     \@tempswafalse
419     \advance\count@\@ne
420     \@for\@e:=\f\do{%
421         \ifnum\@e=\count@
422             \@tempswatrue
423         \fi
424     }%
425     \if@tempswa
426         \expandafter\tabH@do@set
427     \else
428         \expandafter\@gobble
429     \fi
430 }%
431
432 \protected\def\tabH@aux@vstart#1#2#3#4{%
433     \def\tabH@current@vstart{\{#1\}\{#2\}\{#3\}\{#4\}}%
434 }%
435 \protected\def\tabH@aux@vstop{%
436     \expandafter\tabH@aux@v\tabH@current@vstart
437 }%
438 \def\tabH@aux@v#1#2#3#4#5{%
439     \expandafter\gdef\csname tabH@#1row#2x\endcsname{#3}%
440     \expandafter\xdef\csname tabH@#1row#2y\endcsname{%
441         \the\dimexpr #4 - #5\relax
442     }%

```

```

443 }%
444
445 \csname fi\endcsname
446 \endinput
447
448 \fi

```

## 2.5 DVI drivers

```

449 \ifx\tabH@driver\@empty
450 \PackageError{tabularht}{%
451   Missing DVI driver, option 'vlines' disabled%
452 }{%
453   Supported DVI drivers: dvips.%
454 }%
455 \expandafter\endinput
456 \fi
457
458 \def\tabH@driver@dvips{%
459 \def\tabH@literalps##1{\special{ps:SDict begin ##1 end}}}%
460 \def\tabH@headerps##1{\special{! ##1}}}%
461 }
462
463 \@onelevel@sanitize\tabH@driver
464 \@ifundefined{tabH@driver@\tabH@driver}{%
465 \PackageError{tabularht}{%
466   Unsupported driver '\tabH@driver'%
467 }{%
468   Supported DVI drivers: dvips.%
469 }%
470 \endinput
471 }{}
472
473 \begingroup
474 \let\on@line\@empty
475 \PackageInfo{tabularht}{%
476   Using driver '\tabH@driver'%
477 }%
478 \endgroup
479 \csname tabH@driver@\tabH@driver\endcsname
480
481 \protected\def\tabH@vrule#1#2\vrule#3\arrayrulewidth{%
482   #2% \fi or empty
483   % hack to get rid of maxdrift rounding of dvips,
484   % thus simulate a large motion
485   \kern1in\relax
486   \tabH@literalps{%
487     #1 tabH.vrule %
488     Resolution neg 0 translate%
489   }%
490   \vrule#3\arrayrulewidth
491   \tabH@literalps{Resolution 0 translate}%
492   \kern-1in\relax
493 }
494
495 \def\tabH@vspace@start#1{%
496 \begingroup
497 \let\y\@empty
498 \@for\x:=#1\do{%
499   \ifx\y\@empty
500     \edef\y{\x}%
501   \else
502     \edef\y{\y\space\x}%

```

```

503     \fi
504   }%
505   \tabH@literalps{\tabH@currenttab[\y]currentpoint exch pop}%
506 \endgroup
507 }
508 \def\tabH@vspace@stop{%
509   \tabH@literalps{%
510     currentpoint exch pop %
511     \number\dimexpr\arrayrulewidth\relax\space
512     tabH.vspace%
513   }%
514 }
515
516 \tabH@headerps{%
517   userdict begin%
518     /tabH.list 10 dict def%
519     /tabH.job [] def %
520   end%
521   /tabH.vrule{%
522     10 string cvs cvn dup tabH.list exch known{%
523       tabH.list exch dup [ exch tabH.list exch get %
524       currentpoint pop round exch true exch{%
525         % tabH.list key [ ... x true i
526         % tabH.list key [ ... false i
527       exch{%
528         % ... [ ... x i
529         2 copy lt{false}{%
530           2 copy eq{pop false}{exch true}ifelse%
531         }ifelse%
532       }{false}ifelse%
533     }forall %
534     pop%
535     ]put%
536   }{%
537     tabH.list exch[currentpoint pop round]put%
538   }ifelse%
539 }bind def%
540 % <tab num> <cols array> <ytop> <ybottom> <rulewidth[sp]>
541 /tabH.vspace{%
542   userdict begin %
543     10 dict dup begin %
544       exch 65536 div Resolution mul 72.27 div %
545       % dvips uses a poor man's ceil function
546       % see dopage.c before "drawrule": (int)(... + 0.9999999)
547       0.9999999 add truncate%
548       /rulewidth exch def %
549       exch/ybottom exch def %
550       exch/ytop exch def %
551       exch/cols exch def %
552       exch/tabkey exch 10 string cvs cvn def %
553     end%
554     /tabH.job exch[exch userdict/tabH.job get aload pop]def %
555   end%
556 }bind def %
557 % Now we do the work at the end of the page.
558 % Unhappily "eop-hook" cannot be used, because "eop"
559 % executes "restore" before, so that all data are lost.
560 TeXDict begin%
561   /eop%
562   [%
563     {%
564       tabH.job{%

```

```

565     begin%
566     /colarray %
567     tabH.list tabkey known{tabH.list tabkey get}{}{}ifelse %
568     def %
569     cols length 0 eq not{%
570     /colarray[%
571     cols{1 sub %
572     dup 0 lt{pop}{%
573     dup colarray length ge{pop}{%
574     colarray exch get%
575     }ifelse%
576     }ifelse%
577     }forall%
578     ]def%
579     }if %
580     colarray{%
581     % (rulewidth) == rulewidth == % debug
582     Resolution sub %
583     ytop rulewidth ytop ybottom sub v%
584     }forall %
585     end%
586     }forall%
587     % tabH.list{== ==}forall % debug
588     }bind aload pop %
589     TeXDict /eop get aload pop%
590     ]cvx def %
591 end%
592 }
593 </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:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

### 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
```

**Script installation.** Check the directory TDS:scripts/oberdiek/ for scripts that need further installation steps. Package attachfile2 comes with the Perl script pdfatfi.pl that should be installed in such a way that it can be called as pdfatfi. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

### 3.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain- $\TeX$ :

```
tex tabularht.dtx
```

**TDS.** Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

|                                     |               |  |
|-------------------------------------|---------------|--|
| <code>tabularht.sty</code>          | $\rightarrow$ | <code>tex/latex/oberdiek/tabularht.sty</code>          |
| <code>tabularht.pdf</code>          | $\rightarrow$ | <code>doc/latex/oberdiek/tabularht.pdf</code>          |
| <code>tabularht-example1.tex</code> | $\rightarrow$ | <code>doc/latex/oberdiek/tabularht-example1.tex</code> |
| <code>tabularht-example2.tex</code> | $\rightarrow$ | <code>doc/latex/oberdiek/tabularht-example2.tex</code> |
| <code>tabularht.dtx</code>          | $\rightarrow$ | <code>source/latex/oberdiek/tabularht.dtx</code>       |

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 (te $\TeX$ , mik $\TeX$ , ...) relies on file name databases, you must refresh these. For example, te $\TeX$  users run `texhash` or `mktextlsr`.

### 3.5 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk tabularht.pdf unpack_files output .
```

**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 pdf $\LaTeX$ :

```
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`.
- `\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.

## 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; numbers in roman refer to the code lines where the entry is used.

| Symbols                                |  |
|--|--|
| <code>\@@array</code> .....            | <i>72, 73</i>  |
| <code>\@addtopreamble</code> .....     | <i>228</i>   |
| <code>\@addtoreset</code> .....        | <i>212</i>   |
| <code>\@array</code> .....             | <i>58, 68, 72, 73</i>  |
| <code>\@arrayrule</code> .....         | <i>227, 244, 247</i>   |
| <code>\@auxout</code> .....            | <i>275, 366, 402</i>   |
| <code>\@depth</code> .....             | <i>413</i>   |
| <code>\@empty</code> .....             | <i>51, 52, 64,</i><br><i>99, 171, 215, 218, 220, 230, 266,</i><br><i>297, 340, 385, 449, 474, 497, 499</i> |
| <code>\@firstofone</code> .....        | <i>309, 312</i>  |
| <code>\@for</code> .....               | <i>420, 498</i>  |
| <code>\@gobble</code> .....            | <i>170, 251, 312, 317, 428</i>   |
| <code>\@gobblefour</code> .....        | <i>316</i>   |
| <code>\@gobbletwo</code> .....         | <i>212, 315</i>  |
| <code>\@ifpackageloaded</code> .....   | <i>82, 84, 246</i>   |
| <code>\@ifundefined</code> .....       | <i>201, 255, 309, 312, 330, 464</i>  |
| <code>\@mainaux</code> .....           | <i>299, 305, 321</i>   |
| <code>\@nameuse</code> .....           | <i>130, 132</i>  |
| <code>\@one</code> .....               | <i>419</i>   |
| <code>\@nil</code> .....               | <i>56, 68, 92, 94</i>  |
| <code>\@onelevel@sanitize</code> ..... | <i>463</i>   |
| <code>\@percentchar</code> .....       | <i>300, 322</i>  |
| <code>\@secondoftwo</code> .....       | <i>309</i>   |
| <code>\@sharp</code> .....             | <i>56, 65</i>  |
| <code>\@tempswafalse</code> .....      | <i>418</i>   |
| <code>\@tempwattrue</code> .....       | <i>422</i>   |
| <code>\@toarrayheight</code> .....     | <i>51, 62, 64, 97, 116, 218</i>  |
| <code>\@undefined</code> .....         | <i>337, 341, 349</i>   |
| <code>\@width</code> .....             | <i>412</i>   |

|                               |   |                               |  |
|-------------------------------|---|-------------------------------|--|
| <code>\</code>                | 8, 10, 12, 26, 32, 37, 41, 95                                   | <code>\ifx</code>             | 72, 95, 100, 102, 195, 218, 230, 235, 341, 349, 352, 377, 382, 385, 449, 499 |
| <b>A</b>                      |   |                               |  |
| <code>\a</code>               | 375, 376, 379   | <code>\immediate</code>       | 299, 305, 321  |
| <code>\addlinespace</code>    | 29  | <code>\interrowfill</code>    | 2, 157   |
| <code>\advance</code>         | 419   | <code>\interrowsspace</code>  | 2, 39, 149   |
| <code>\array</code>           | 144   | <code>\interrowstart</code>   | 2, 28, 33, 173   |
| <code>\arrayrulewidth</code>  | 229, 241, 412, 481, 490, 511                                    | <code>\interrowstop</code>    | 30, 36, 181  |
| <code>\AtBeginDocument</code> | 83, 245, 291, 293   | <b>K</b>                      |  |
| <b>B</b>                      |   |                               |  |
| <code>\begin</code>           | 5, 7, 22, 24, 304   | <code>\kern</code>            | 485, 492   |
| <b>C</b>                      |   |                               |  |
| <code>\c@tabH@unique</code>   | 223, 364  | <b>L</b>                      |  |
| <code>\copy</code>            | 394   | <code>\l</code>               | 380, 382, 391  |
| <code>\count@</code>          | 388, 419, 421   | <b>M</b>                      |  |
| <code>\csname</code>          | 235, 262, 331, 339, 340, 352, 376, 379, 381, 439, 440, 445, 479 | <code>\meaning</code>         | 106  |
| <code>\CT@arc@</code>         | 237   | <code>\MessageBreak</code>    | 109  |
| <b>D</b>                      |   |                               |  |
| <code>\DeclareOption</code>   | 186, 189, 190   | <code>\multicolumn</code>     | 10   |
| <code>\detokenize</code>      | 306   | <b>N</b>                      |  |
| <code>\dimen@</code>          | 96, 97, 115, 116  | <code>\NeedsTeXFormat</code>  | 48   |
| <code>\dimexpr</code>         | 411, 413, 511   | <code>\newcommand</code>      | 153, 161, 173, 177, 181  |
| <code>\do</code>              | 332, 338, 342, 354, 358, 386, 389, 420, 498                     | <code>\newcounter</code>      | 213  |
| <code>\documentclass</code>   | 2, 18   | <code>\newenvironment</code>  | 121, 128, 135, 142   |
| <code>\dp</code>              | 393   | <code>\noalign</code>         | 9, 11, 150, 158, 174, 182  |
| <b>E</b>                      |   |                               |  |
| <code>\e</code>               | 420, 421  | <code>\number</code>          | 511  |
| <code>\end</code>             | 13, 15, 43, 45, 293   | <code>\numexpr</code>         | 441  |
| <code>\endarray</code>        | 146   | <b>O</b>                      |  |
| <code>\endcsname</code>       | 235, 262, 331, 339, 340, 352, 376, 379, 381, 439, 440, 445, 479 | <code>\on@line</code>         | 266, 474   |
| <code>\endinput</code>        | 197, 208, 263, 446, 455, 470                                    | <b>P</b>                      |  |
| <code>\endtabular</code>      | 125   | <code>\PackageError</code>    | 104, 202, 256, 450, 465  |
| <code>\endtabularx</code>     | 139   | <code>\PackageInfo</code>     | 267, 475   |
| <code>\extracolsep</code>     | 7   | <code>\pdfastxpos</code>      | 283  |
| <b>F</b>                      |   |                               |  |
| <code>\f</code>               | 384, 385, 420   | <code>\pdfastypos</code>      | 284  |
| <code>\fbox</code>            | 6   | <code>\pdfsavepos</code>      | 259, 274, 365, 401   |
| <code>\fill</code>            | 7, 162  | <code>\ProcessOptions</code>  | 193  |
| <b>G</b>                      |   |                               |  |
| <code>\gdef</code>            | 58, 439   | <code>\protected</code>       | 272, 329, 432, 435, 481  |
| <b>H</b>                      |   |                               |  |
| <code>\hbox</code>            | 391, 410  | <code>\protected@write</code> | 275, 366, 402  |
| <code>\hline</code>           | 25, 27, 31, 34, 35, 38, 40, 42                                  | <code>\providecommand</code>  | 184  |
| <code>\hskip</code>           | 229, 241, 411   | <code>\ProvidesPackage</code> | 49   |
| <code>\hss</code>             | 414   | <b>R</b>                      |  |
| <b>I</b>                      |   |                               |  |
| <code>\if@filesw</code>       | 273, 298, 362, 400  | <code>\renewcommand</code>    | 227  |
| <code>\if@tempswa</code>      | 425   | <code>\RequirePackage</code>  | 252  |
| <code>\ifnum</code>           | 350, 353, 421   | <code>\reserved@a</code>      | 73, 75, 78   |
| <code>\ifpdf</code>           | 253   | <b>S</b>                      |  |
| <b>T</b>                      |   |                               |  |
| <code>\tabH@</code>           | 328   | <code>\setbox</code>          | 391  |
| <code>\tabH@@setheight</code> | 92, 94  | <code>\setlength</code>       | 96, 115  |
| <code>\tabH@array@init</code> | 52, 59, 217   | <code>\space</code>           | 259, 350, 353, 502, 511  |
| <code>\tabH@arrayrule</code>  | 244, 247  | <code>\special</code>         | 459, 460   |
| <code>\tabH@aux@</code>       | 296, 307  | <code>\stepcounter</code>     | 222, 363   |
| <code>\tabH@aux@done</code>   | 294, 297  | <code>\strip@prefix</code>    | 106  |



