

# The `tabularht` package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2006/12/22 v2.3

## 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	13
3.4	Refresh file name databases	14
3.5	Some details for the interested	14
<b>4</b>	<b>History</b>	<b>14</b>
[2005/09/22 v1.0]		14
[2005/10/16 v2.0]		15
[2005/10/18 v2.1]		15
[2006/02/20 v2.2]		15
[2006/12/22 v2.3]		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=`.

```
\interrowspace [<list>] {<height>}
\interrowfill  [<list>]
\interrowstart [<list>] ... \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 \usepackage{tabularht}
3
4 \begin{document}
5 \fbox{%
6   \begin{tabularht*}{1in}{4in}{@{}l@{\extracolsep{\fill}}r@{}}
7     upper left corner & upper right corner\\
8     \noalign{\vfill}%
9     \multicolumn{2}{@{}c@{}}{bounding box}\\
10    \noalign{\vfill}%
11    lower left corner & lower right corner\\
12    \end{tabularht*}%
13 }
14 \end{document}
15 \end{document}
16 \end{document}
```

### 1.4.2 Example 2

```
17 \documentclass{article}
18 \usepackage{booktabs}
19 \usepackage{dvips,vlines}{tabularht}
20
21 \begin{document}
22 \begin{tabularht}{spread=0pt}{|l|l|}
23   \hline
24   First&Line\\
25 \end{tabularht}
```

```

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 [2006/12/22 v2.3 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\vcenter\fi\fi\bgroup#2\@sharp#3#4\@nil{%
57       \endgroup
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
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\t@arrayheight{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\t@arrayheight{\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\interrowospace{%
150 \noalign\bgroup
151 \tabH@interrowospace
152 }
153 \newcommand*{\tabH@interrowospace}[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 \newcounter{tabH@unique}
212 \setcounter{tabH@unique}{0}
213 \let\tabH@currenttab\@empty
214
215 \def\tabH@array@init{%
216   \ifx\@toarrayheight\@empty
217     % ignore vertical lines of nested tabular environments
218     \let\tabH@currenttab\@empty
219   \else
220     \stepcounter{tabH@unique}%
221     \edef\tabH@currenttab{\the\c@tabH@unique}%
222   \fi
223 }
224
225 \renewcommand*{\@arrayrule}{%
226   \@addtopreamble{%
227     \hskip -.5\arrayrulewidth
228     \ifx\tabH@currenttab\@empty
229     \else
230       \tabH@vrule{\tabH@currenttab}%
231     \fi
232     \begingroup
233       \expandafter\ifx\csname CT@arc@\endcsname\relax
234       \else
235         \expandafter\CT@arc@
236       \fi
237       \vline
238     \endgroup
239     \hskip -.5\arrayrulewidth
240   }%
241 }
242 \let\tabH@arrayrule\@arrayrule
243 \AtBeginDocument{%
244   \@ifpackageloaded{colortbl}{%
245     \let\@arrayrule\tabH@arrayrule
246   }{}%
247 }
248
249 \let\tabH@vrule\@gobble

```

## 2.4 Driver pdftex

```

250 \RequirePackage{ifpdf}
251 \ifpdf
252   \begingroup
253     \@ifundefined{pdfsavepos}{%
254       \PackageError{tabularht}{%
255         Your pdfTeX is too old%
256       }{%
257         \string\pdfsavepos\space is missing.%
258       }%
259     \endgroup
260     \csname fi\endcsname

```

```

261     \endinput
262 }{}%
263
264     \let\on@line\@empty
265     \PackageInfo{tabularht}{%
266         Using driver 'pdftex' because of pdfTeX in PDF mode%
267     }%
268 \endgroup
269
270 \protected\def\tabH@vrule#1{%
271     \if@filesw
272         \pdfsavepos
273         \protected@write\@auxout{%
274             \let\tabH@lastxpos\relax
275         }{%
276             \tabH@aux@vrule{#1}{\tabH@lastxpos}%
277         }%
278     \fi
279 }
280
281 \def\tabH@lastxpos{\the\pdfastxpos}
282 \def\tabH@lastypos{\the\pdfastypos}
283
284 % The .aux file contains three commands:
285 % \tabH@aux@vrule{tabular id}{x position}
286 % \tabH@aux@vstart{tabular id}{row id}{x position}{y position}
287 % \tabH@aux@vstop{y position}
288 %
289 \AtBeginDocument{%
290     % The .aux files are read the first time before
291     % \AtBeginDocument and later at \end{document}.
292     % \tabH@aux@done is a marker to distinguish
293     % between these two readings. Only in the first
294     % case we need the \tabH@aux@... commands.
295     \let\tabH@aux@done\@empty
296     \if@filesw
297         \immediate\write\@mainaux{%
298             \@percentchar\@percentchar BeginProlog: tabularht
299         }%
300         % items in the aux file are executed,
301         % if tabularht is loaded
302         % and during the aux file read at \begin{document} only
303         \immediate\write\@mainaux{%
304             \detokenize{%
305                 % the \tabH@aux@... commands are needed only if
306                 % tabularht is loaded with driver pdftex.
307                 \@ifundefined{tabH@aux@vrule}{\@secondoftwo}{\@firstofone}
308                 {%
309                     % disable commands except for the first .aux files reading
310                     \@ifundefined{tabH@aux@done}{\@gobble}{\@firstofone}
311                 }%
312                 {%
313                     \let\tabH@aux@vrule\@gobbletwo
314                     \let\tabH@aux@vstart\@gobblefour
315                     \let\tabH@aux@vstop\@gobble
316                 }%
317             }%
318         }%
319         \immediate\write\@mainaux{%
320             \@percentchar\@percentchar EndProlog: tabularht
321         }%
322     \fi

```



```

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

```

```

385         \else
386         \count@=\z@
387         \let\do\tabH@do@filter
388         \fi
389         \setbox\z@=\hbox{\l}%
390         \wd\z@=\z@
391         \dp\z@=\z@
392         \copy\z@
393     \fi
394 \fi
395 \endgroup
396 }%
397 \def\tabH@vspace@stop{%
398     \if@filesw
399     \pdfsavepos
400     \protected@write\@auxout{%
401         \let\tabH@lastypos\relax
402     }{%
403         \tabH@aux@vstop{\tabH@lastypos}%
404     }%
405 \fi
406 }
407 \def\tabH@do@set#1{%
408     \hbox to \z@{%
409         \hskip \dimexpr #1sp - \x sp\relax
410         \vrule \@width\arrayrulewidth
411             \@depth\dimexpr \y sp\relax
412     \hss
413     }%
414 }
415 \def\tabH@do@filter{%
416     \@tempswafalse
417     \advance\count@\@ne
418     \@for\@e:=\f\do{%
419         \ifnum\@e=\count@
420             \@tempwattrue
421         \fi
422     }%
423     \if@tempswa
424         \expandafter\tabH@do@set
425     \else
426         \expandafter\@gobble
427     \fi
428 }
429
430 \protected\def\tabH@aux@vstart#1#2#3#4{%
431     \def\tabH@current@vstart{{#1}{#2}{#3}{#4}}%
432 }
433 \protected\def\tabH@aux@vstop{%
434     \expandafter\tabH@aux@v\tabH@current@vstart
435 }
436 \def\tabH@aux@v#1#2#3#4#5{%
437     \expandafter\gdef\csname tabH@#1row#2x\endcsname{#3}%
438     \expandafter\xdef\csname tabH@#1row#2y\endcsname{%
439         \the\dimexpr #4 - #5\relax
440     }%
441 }
442
443 \csname fi\endcsname
444 \endinput
445
446 \fi

```

## 2.5 DVI drivers

```

447 \ifx\tabH@driver\@empty
448   \PackageError{tabularht}{%
449     Missing DVI driver, option 'vlines' disabled%
450   }{%
451     Supported DVI drivers: dvips.%
452   }%
453   \expandafter\endinput
454 \fi
455
456 \def\tabH@driver@dvips{%
457   \def\tabH@literalps##1{\special{ps:SDict begin ##1 end}}}%
458   \def\tabH@headerps##1{\special{! ##1}}}%
459 }
460
461 \@onelevel@sanitize\tabH@driver
462 \ifundefined{tabH@driver@\tabH@driver}{%
463   \PackageError{tabularht}{%
464     Unsupported driver '\tabH@driver'%
465   }{%
466     Supported DVI drivers: dvips.%
467   }%
468   \endinput
469 }{}
470
471 \begingroup
472   \let\on@line\@empty
473   \PackageInfo{tabularht}{%
474     Using driver '\tabH@driver'%
475   }%
476 \endgroup
477 \csname tabH@driver@\tabH@driver\endcsname
478
479 \protected\def\tabH@vrule#1#2\vrule#3\arrayrulewidth{%
480   #2% \fi or empty
481   % hack to get rid of maxdrift rounding of dvips,
482   % thus simulate a large motion
483   \kern1in\relax
484   \tabH@literalps{%
485     #1 tabH.vrule
486     Resolution neg 0 translate%
487   }%
488   \vrule#3\arrayrulewidth
489   \tabH@literalps{Resolution 0 translate}%
490   \kern-1in\relax
491 }
492
493 \def\tabH@vspace@start#1{%
494   \begingroup
495     \let\y\@empty
496     \@for\x:=#1\do{%
497       \ifx\y\@empty
498         \edef\y{\x}%
499       \else
500         \edef\y{\y\space\x}%
501       \fi
502     }%
503     \tabH@literalps{\tabH@currenttab[\y]currentpoint exch pop}%
504   \endgroup
505 }
506 \def\tabH@vspace@stop{%
507   \tabH@literalps{%

```

```

508     currentpoint exch pop %
509     \number\dimexpr\arrayrulewidth\relax\space
510     tabH.vspace%
511 }%
512 }
513
514 \tabH@headerps{%
515   userdict begin
516     /tabH.list 10 dict def
517     /tabH.job [] def
518   end%
519   /tabH.vrule{%
520     10 string cvs cvn dup tabH.list exch known{%
521       tabH.list exch dup [ exch tabH.list exch get
522       currentpoint pop round exch true exch{%
523         % tabH.list key [ ... x true i
524         % tabH.list key [ ... false i
525         exch{%
526           % ... [ ... x i
527           2 copy lt{false}{%
528             2 copy eq{pop false}{exch true}ifelse%
529           }ifelse
530         }{false}ifelse
531       }forall
532       pop%
533       ]put%
534     }{%
535       tabH.list exch[currentpoint pop round]put
536     }ifelse
537   }bind def
538   % <tab num> <cols array> <ytop> <ybottom> <rulewidth[sp]>
539   /tabH.vspace{%
540     userdict begin
541       10 dict dup begin
542         exch 65536 div Resolution mul 72.27 div
543         % dvips uses a poor man's ceil function
544         % see dopage.c before "drawrule": (int)(... + 0.9999999)
545         0.9999999 add truncate%
546         /rulewidth exch def
547         exch/ybottom exch def
548         exch/ytop exch def
549         exch/cols exch def
550         exch/tabkey exch 10 string cvs cvn def
551       end
552       /tabH.job exch[exch userdict/tabH.job get aload pop]def
553     end%
554   }bind def
555   % Now we do the work at the end of the page.
556   % Unhappily "eop-hook" cannot be used, because "eop"
557   % executes "restore" before, so that all data are lost.
558   TeXDict begin%
559   /eop%
560   [%
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%
573             }ifelse%
574             }ifelse%
575             }forall%
576         ]def%
577     }if
578     colarray{%
579         % (rulewidth) == rulewidth == % debug
580         Resolution sub
581         ytop rulewidth ytop ybottom sub v
582     }forall
583     end
584 }forall
585 % tabH.list{== ==}forall % debug
586 }bind aload pop
587 TeXDict /eop get aload pop
588 ]cvx def
589 end%
590 }
591 </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/
```

### 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
```

---

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

**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>	→	<code>tex/latex/oberdiek/tabularht.sty</code>
<code>tabularht.pdf</code>	→	<code>doc/latex/oberdiek/tabularht.pdf</code>
<code>tabularht-example1.tex</code>	→	<code>doc/latex/oberdiek/tabularht-example1.tex</code>
<code>tabularht-example2.tex</code>	→	<code>doc/latex/oberdiek/tabularht-example2.tex</code>
<code>tabularht.dtx</code>	→	<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  $\text{\TeX}$  distribution (`te $\text{\TeX}$` , `mik $\text{\TeX}$` , ...) relies on file name databases, you must refresh these. For example, `te $\text{\TeX}$`  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  $\text{\LaTeX}$ .** 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`.
- `\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`.

## 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> .....	72, 73
<code>\@addtopreamble</code> .....	226
<code>\@array</code> .....	58, 68, 72, 73
<code>\@arrayrule</code> .....	225, 242, 245
<code>\@auxout</code> .....	273, 364, 400
<code>\@depth</code> .....	411
<code>\@empty</code> .....	51, 52, 64, 99, 171, 213, 216, 218, 228, 264, 295, 338, 383, 447, 472, 495, 497
<code>\@firstofone</code> .....	307, 310
<code>\@for</code> .....	418, 496
<code>\@gobble</code> .....	170, 249, 310, 315, 426
<code>\@gobblefour</code> .....	314
<code>\@gobbletwo</code> .....	313
<code>\@ifpackageloaded</code> .....	82, 84, 244
<code>\@ifundefined</code> .....	201, 253, 307, 310, 328, 462
<code>\@mainaux</code> .....	297, 303, 319
<code>\@nameuse</code> .....	130, 132
<code>\@ne</code> .....	417
<code>\@nil</code> .....	56, 68, 92, 94
<code>\@onelevel@sanitize</code> .....	461
<code>\@percentchar</code> .....	298, 320
<code>\@secondoftwo</code> .....	307
<code>\@sharp</code> .....	56, 65
<code>\@tempswafalse</code> .....	416
<code>\@tempwattrue</code> .....	420
<code>\@toarrayheight</code> .....	51, 62, 64, 97, 116, 216
<code>\@undefined</code> .....	335, 339, 347
<code>\@width</code> .....	410
<code>\@</code> .....	8, 10, 12, 26, 32, 37, 41, 95
A	
<code>\a</code> .....	373, 374, 377
<code>\addlinespace</code> .....	29
<code>\advance</code> .....	417
<code>\array</code> .....	144
<code>\arrayrulewidth</code> .....	227, 239, 410, 479, 488, 509
<code>\AtBeginDocument</code> .....	83, 243, 289, 291
B	
<code>\begin</code> .....	5, 7, 22, 24, 302
C	
<code>\c@tabH@unique</code> .....	221, 362
<code>\copy</code> .....	392
<code>\count@</code> .....	386, 417, 419
<code>\csname</code> .....	233, 260, 329, 337, 338, 350, 374, 377, 379, 437, 438, 443, 477
<code>\CT@arc@</code> .....	235
D	
<code>\DeclareOption</code> .....	186, 189, 190
<code>\detokenize</code> .....	304
<code>\dimen@</code> .....	96, 97, 115, 116
<code>\dimexpr</code> .....	409, 411, 509
<code>\do</code> .....	330, 336, 340, 352, 356, 384, 387, 418, 496
<code>\documentclass</code> .....	2, 18

<code>\dp</code> .....	391		<b>P</b>	
		<b>E</b>	<code>\PackageError</code> .	104, 202, 254, 448, 463
<code>\e</code> .....	418, 419		<code>\PackageInfo</code> .....	265, 473
<code>\end</code> .....	13, 15, 43, 45, 291		<code>\pdfastxpos</code> .....	281
<code>\endarray</code> .....	146		<code>\pdfastypos</code> .....	282
<code>\endcsname</code> .....			<code>\pdfsavepos</code> .....	257, 272, 363, 399
	233, 260, 329, 337, 338, 350,		<code>\ProcessOptions</code> .....	193
	374, 377, 379, 437, 438, 443, 477		<code>\protected</code> ...	270, 327, 430, 433, 479
<code>\endinput</code> .	197, 208, 261, 444, 453, 468		<code>\protected@write</code> .....	273, 364, 400
<code>\endtabular</code> .....	125		<code>\providecommand</code> .....	184
<code>\endtabularx</code> .....	139		<code>\ProvidesPackage</code> .....	49
<code>\extracolsep</code> .....	7		<b>R</b>	
		<b>F</b>	<code>\renewcommand</code> .....	225
<code>\f</code> .....	382, 383, 418		<code>\RequirePackage</code> .....	250
<code>\fbox</code> .....	6		<code>\reserved@a</code> .....	73, 75, 78
<code>\fill</code> .....	7, 162		<b>S</b>	
		<b>G</b>	<code>\setbox</code> .....	389
<code>\gdef</code> .....	58, 437		<code>\setcounter</code> .....	212
		<b>H</b>	<code>\setlength</code> .....	96, 115
<code>\hbox</code> .....	389, 408		<code>\space</code> .....	257, 348, 351, 500, 509
<code>\hline</code> ....	25, 27, 31, 34, 35, 38, 40, 42		<code>\special</code> .....	457, 458
<code>\hskip</code> .....	227, 239, 409		<code>\stepcounter</code> .....	220, 361
<code>\hss</code> .....	412		<code>\strip@prefix</code> .....	106
		<b>I</b>	<b>T</b>	
<code>\if@filesw</code> .....	271, 296, 360, 398		<code>\tabH@</code> .....	326
<code>\if@tempw</code> .....	423		<code>\tabH@@setheight</code> .....	92, 94
<code>\ifnum</code> .....	348, 351, 419		<code>\tabH@array@init</code> .....	52, 59, 215
<code>\ifpdf</code> .....	251		<code>\tabH@arrayrule</code> .....	242, 245
<code>\ifx</code> .....	72, 95, 100,		<code>\tabH@aux@</code> .....	294, 305
	102, 195, 216, 228, 233, 339,		<code>\tabH@aux@done</code> .....	292, 295
	347, 350, 375, 380, 383, 447, 497		<code>\tabH@aux@v</code> .....	434, 436
<code>\immediate</code> .....	297, 303, 319		<code>\tabH@aux@vrule</code> ...	276, 285, 313, 327
<code>\interrowfill</code> .....	2, 157		<code>\tabH@aux@vstart</code> ..	286, 314, 368, 430
<code>\interrowsspace</code> .....	2, 39, 149		<code>\tabH@aux@vstop</code> ...	287, 315, 403, 433
<code>\interrowstart</code> .....	2, 28, 33, 173		<code>\tabH@current@vstart</code> .....	431, 434
<code>\interrowstop</code> .....	30, 36, 181		<code>\tabH@currentrow</code> .....	362, 368, 373
		<b>K</b>	<code>\tabH@currenttab</code> .....	213, 218,
<code>\kern</code> .....	483, 490			221, 228, 230, 368, 373, 379, 503
		<b>L</b>	<code>\tabH@do@add</code> .....	336, 346
<code>\l</code> .....	378, 380, 389		<code>\tabH@do@filter</code> .....	387, 415
		<b>M</b>	<code>\tabH@do@set</code> .....	384, 407, 424
<code>\meaning</code> .....	106		<code>\tabH@driver</code> .....	184,
<code>\MessageBreak</code> .....	109			191, 447, 461, 462, 464, 474, 477
<code>\multicolumn</code> .....	10		<code>\tabH@driver@dvi</code> .....	456
		<b>N</b>	<code>\tabH@headerps</code> .....	458, 514
<code>\NeedsTeXFormat</code> .....	48		<code>\tabH@interrowfill</code> .....	159, 161
<code>\newcommand</code> ...	153, 161, 173, 177, 181		<code>\tabH@interrowsspace</code> .....	151, 153
<code>\newcounter</code> .....	211		<code>\tabH@interrowstart</code> .....	175, 177
<code>\newenvironment</code> ...	121, 128, 135, 142		<code>\tabH@lastxpos</code> .....	274, 276, 281, 365, 369
<code>\noalign</code> ....	9, 11, 150, 158, 174, 182		<code>\tabH@lastypos</code> .....	282, 366, 369, 401, 403
<code>\number</code> .....	509		<code>\tabH@literalps</code> .....	457, 484, 489, 503, 507
<code>\numexpr</code> .....	439		<code>\tabH@patch@array</code> .....	71, 80, 85
		<b>O</b>	<code>\tabH@patch@array</code> .....	70, 77
<code>\on@line</code> .....	264, 472		<code>\tabH@setheight</code> .....	90, 122, 129, 136, 143
			<code>\tabH@spread</code> .....	102, 120
			<code>\tabH@temp</code> .....	99,
				100, 102, 106, 112, 116, 187, 195
			<code>\tabH@to</code> .....	100, 112, 119
			<code>\tabH@vrule</code> .....	230, 249, 270, 479
			<code>\tabH@vspace</code> .....	154, 162, 165



<code>\tabH@vspace@start</code> .....	<code>\vspace</code> .....	167
..... 166, 170, 178, 359, 493		
<code>\tabH@vspace@stop</code> .....	<b>W</b>	
..... 168, 171, 182, 397, 506	<code>\wd</code> .....	390
<code>\tabular</code> .....	<code>\write</code> .....	297, 303, 319
123		
<code>\tabularx</code> .....	<b>X</b>	
137		
<code>\the</code> . 70, 97, 116, 221, 281, 282, 362, 439	<code>\x</code> .....	56, 68, 334, 340, 348, 351,
<code>\toks@</code> .....	..... 352, 374, 375, 409, 496, 498, 500	
54, 70		
<code>\tracingmacros</code> .....		
91		
<b>U</b>		
<code>\usepackage</code> .....	<b>Y</b>	
3, 19, 20	<code>\y</code> .....	335, 339, 347,
<b>V</b>		
<code>\vcenter</code> .....	..... 377, 411, 495, 497, 498, 500, 503	
56, 61		
<code>\vfill</code> .....	<b>Z</b>	
9, 11		
<code>\vline</code> .....	<code>\z@</code> .....	386, 389, 390, 391, 392, 408
237	<code>\zap@space</code> .....	99
<code>\vrule</code> .....		
410, 479, 488		