

The `tabularht` package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2006/02/20 v2.2

Abstract

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

Contents

1	Usage	1
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
2	Implementation	4
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	10
3	Installation	13
3.1	Some details for the interested	13
4	History	14
[2005/09/22 v1.0]		14
[2005/10/16 v2.0]		14
[2005/10/18 v2.1]		14
[2006/02/20 v2.2]		14
5	Index	14

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`
- `tabularxht`, `tabularxht*` (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 ε -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_EX: 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    lower left corner & lower right corner\\
13  \end{tabularht*}%
14 }
15 \end{document}
16 \end{example1}
```

1.4.2 Example 2

```

17 \documentclass{article}
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 [2006/02/20 v2.2 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         \@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\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 \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       \csname CT@arc@\endcsname\vline
234     \endgroup
235     \hskip -.5\arrayrulewidth
236   }%
237 }
238 \let\tabH@arrayrule\@arrayrule
239 \AtBeginDocument{%
240   \@ifpackageloaded{colortbl}{%
241     \let\@arrayrule\tabH@arrayrule
242   }{}%
243 }
244
245 \let\tabH@vrule\@gobble

```

2.4 Driver pdftex

```

246 \RequirePackage{ifpdf}
247 \ifpdf
248   \begingroup
249     \@ifundefined{pdfsavepos}{%
250       \PackageError{tabularht}{%
251         Your pdfTeX is too old%
252       }{%
253         \string\pdfsavepos\space is missing.%
254       }%
255     \endgroup
256     \csname fi\endcsname
257     \endinput
258   }{}%
259
260   \let\on@line\@empty
261   \PackageInfo{tabularht}{%
262     Using driver 'pdftex' because of pdfTeX in PDF mode%

```

```

263     }%
264 \endgroup
265
266 \protected\def\tabH@vrule#1{%
267   \if@filesw
268     \pdfsavepos
269     \protected@write\@auxout{%
270       \let\tabH@lastxpos\relax
271     }{%
272       \tabH@aux@vrule{#1}{\tabH@lastxpos}%
273     }%
274   \fi
275 }
276
277 \def\tabH@lastxpos{\the\pdflastxpos}
278 \def\tabH@lastypos{\the\pdflastypos}
279
280 % The .aux file contains three commands:
281 % \tabH@aux@vrule{tabular id}{x position}
282 % \tabH@aux@vstart{tabular id}{row id}{x position}{y position}
283 % \tabH@aux@vstop{y position}
284 %
285 \AtBeginDocument{%
286   % The .aux files are read the first time before
287   % \AtBeginDocument and later at \end{document}.
288   % \tabH@aux@done is a marker to distinguish
289   % between these two readings. Only in the first
290   % case we need the \tabH@aux@... commands.
291   \let\tabH@aux@done\@empty
292   \if@filesw
293     \immediate\write\@mainaux{%
294       \@percentchar\@percentchar BeginProlog: tabularht
295     }%
296     % items in the aux file are executed,
297     % if tabularht is loaded
298     % and during the aux file read at \begin{document} only
299     \immediate\write\@mainaux{%
300       \detokenize{%
301         % the \tabH@aux@... commands are needed only if
302         % tabularht is loaded with driver pdftex.
303         \@ifundefined{tabH@aux@vrule}\@secondoftwo\@firstofone
304       }{%
305         % disable commands except for the first .aux files reading
306         \@ifundefined{tabH@aux@done}\@gobble\@firstofone
307       }%
308       {%
309         \let\tabH@aux@vrule\@gobbletwo
310         \let\tabH@aux@vstart\@gobblefour
311         \let\tabH@aux@vstop\@gobble
312       }%
313     }%
314   }%
315   \immediate\write\@mainaux{%
316     \@percentchar\@percentchar EndProlog: tabularht
317   }%
318 \fi
319 }
320
321 % the x positions of vrules are stored in
322 % \tabH@<tabcount>list with distinct values
323 \protected\def\tabH@aux@vrule#1#2{%
324   \@ifundefined{tabH@#1list}{%

```

```

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

```

```

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

```

2.5 DVI drivers

```

443 \ifx\tabH@driver\@empty
444     \PackageError{tabularht}{%
445         Missing DVI driver, option ‘vlines’ disabled%
446     }{%

```

```

447     Supported DVI drivers: dvips.%
448 }%
449 \expandafter\endinput
450 \fi
451
452 \def\tabH@driver@dvips{%
453   \def\tabH@literalps##1{\special{ps:SDict begin ##1 end}}}%
454   \def\tabH@headerps##1{\special{! ##1}}}%
455 }
456
457 \@onelevel@sanitize\tabH@driver
458 \@ifundefined{tabH@driver@\tabH@driver}{%
459   \PackageError{tabularht}{%
460     Unsupported driver '\tabH@driver'%
461   }{%
462     Supported DVI drivers: dvips.%
463   }%
464   \endinput
465 }{}
466
467 \begingroup
468   \let\on@line\@empty
469   \PackageInfo{tabularht}{%
470     Using driver '\tabH@driver'%
471   }%
472 \endgroup
473 \csname tabH@driver@\tabH@driver\endcsname
474
475 \protected\def\tabH@vrule#1#2\vrule#3\arrayrulewidth{%
476   #2% \fi or empty
477   % hack to get rid of maxdrift rounding of dvips,
478   % thus simulate a large motion
479   \kern1in\relax
480   \tabH@literalps{%
481     #1 tabH.vrule
482     Resolution neg 0 translate%
483   }%
484   \vrule#3\arrayrulewidth
485   \tabH@literalps{Resolution 0 translate}%
486   \kern-1in\relax
487 }
488
489 \def\tabH@vspace@start#1{%
490   \begingroup
491     \let\y\@empty
492     \@for\x:=#1\do{%
493       \ifx\y\@empty
494         \edef\y{\x}%
495       \else
496         \edef\y{\y\space\x}%
497       \fi
498     }%
499     \tabH@literalps{\tabH@currenttab[\y]currentpoint exch pop}%
500   \endgroup
501 }
502 \def\tabH@vspace@stop{%
503   \tabH@literalps{%
504     currentpoint exch pop %
505     \number\dimexpr\arrayrulewidth\relax\space
506     tabH.vspace%
507   }%
508 }

```

```

509
510 \tabH@headerps{%
511   userdict begin
512     /tabH.list 10 dict def
513     /tabH.job [] def
514   end%
515 /tabH.vrule{%
516   10 string cvs cvn dup tabH.list exch known{%
517     tabH.list exch dup [ exch tabH.list exch get
518     currentpoint pop round exch true exch{%
519       % tabH.list key [ ... x true i
520       % tabH.list key [ ... false i
521     exch{%
522       % ... [ ... x i
523       2 copy lt{false}{%
524         2 copy eq{pop false}{exch true}ifelse%
525     }ifelse
526   }{false}ifelse
527 }forall
528 pop%
529 ]put%
530 }{%
531   tabH.list exch[currentpoint pop round]put
532 }ifelse
533 }bind def
534 % <tab num> <cols array> <ytop> <ybottom> <rulewidth[sp]>
535 /tabH.vspace{%
536   userdict begin
537     10 dict dup begin
538       exch 65536 div Resolution mul 72.27 div
539       % dvips uses a poor man's ceil function
540       % see dopage.c before "drawrule": (int)(... + 0.9999999)
541       0.9999999 add truncate%
542     /rulewidth exch def
543     exch/ybottom exch def
544     exch/ytop exch def
545     exch/cols exch def
546     exch/tabkey exch 10 string cvs cvn def
547   end
548   /tabH.job exch[exch userdict/tabH.job get aload pop]def
549   end%
550 }bind def
551 % Now we do the work at the end of the page.
552 % Unhappily "eop-hook" cannot be used, because "eop"
553 % executes "restore" before, so that all data are lost.
554 TeXDict begin%
555 /eop%
556 [%
557   {%
558     tabH.job{%
559       begin%
560         /colarray
561         tabH.list tabkey known{tabH.list tabkey get}{}ifelse
562         def
563         cols length 0 eq not{%
564           /colarray[%
565             cols{1 sub
566               dup 0 lt{pop}{%
567                 dup colarray length ge{pop}{%
568                   colarray exch get%
569                 }ifelse%
570               }ifelse%

```

```

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

```

3 Installation

CTAN. This package is available on CTAN¹:

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

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

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):

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.

Refresh file databases. If your TeX distribution (TeX, MiKTeX, ...) rely on file databases, you must refresh these. For example, TeX users run `texhash` or `mktextlsr`.

3.1 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 .
```

¹<http://ftp.ctan.org/tex-archive/>

Unpacking with L^AT_EX. The .dtx chooses its action depending on the format:

plain-T_EX: Run docstrip and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for docstrip (really, docstrip does not need L^AT_EX), then inform the autodetect routine about your intension:

```
latex \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 pdfL^AT_EX:

```
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.
- \interrowospace, \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.

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	\@addtopreamble	226
\@@array	72, 73	\@array 58, 68, 72, 73

\pdfsavepos	253, 268, 359, 395	\tabH@lastxpos	270, 272, 277, 361, 365
\ProcessOptions	193	\tabH@lastypos	278, 362, 365, 397, 399
\protected ...	266, 323, 426, 429, 475	\tabH@literalps	453, 480, 485, 499, 503
\protected@write	269, 360, 396	\tabH@patch@@array	71, 80, 85
\providecommand	184	\tabH@patch@array	70, 77
\ProvidesPackage	49	\tabH@setheight	90, 122, 129, 136, 143
R		\tabH@spread	102, 120
\renewcommand	225	\tabH@temp	99,
\RequirePackage	246		100, 102, 106, 112, 116, 187, 195
\reserved@a	73, 75, 78	\tabH@to	100, 112, 119
S		\tabH@vrule	230, 245, 266, 475
\setbox	385	\tabH@vspace	154, 162, 165
\setcounter	212	\tabH@vspace@start	
\setlength	96, 115		166, 170, 178, 355, 489
\space	253, 344, 347, 496, 505	\tabH@vspace@stop	
\special	453, 454		168, 171, 182, 393, 502
\stepcounter	220, 357	\tabular	123
\strip@prefix	106	\tabularx	137
T		\the	70, 97, 116, 221, 277, 278, 358, 435
\tabH@	322	\toks@	54, 70
\tabH@@setheight	92, 94	\tracingmacros	91
\tabH@array@init	52, 59, 215	U	
\tabH@arrayrule	238, 241	\usepackage	3, 19, 20
\tabH@aux@	290, 301	V	
\tabH@aux@done	288, 291	\vcenter	56, 61
\tabH@aux@v	430, 432	\vfill	9, 11
\tabH@aux@vrule ...	272, 281, 309, 323	\vline	233
\tabH@aux@vstart ..	282, 310, 364, 426	\vrule	406, 475, 484
\tabH@aux@vstop ...	283, 311, 399, 429	\vspace	167
\tabH@current@vstart	427, 430	W	
\tabH@currentrow	358, 364, 369	\wd	386
\tabH@currenttab	213, 218,	\write	293, 299, 315
	221, 228, 230, 364, 369, 375, 499	X	
\tabH@do@add	332, 342	\x	56, 68, 330, 336, 344, 347,
\tabH@do@filter	383, 411		348, 370, 371, 405, 492, 494, 496
\tabH@do@set	380, 403, 420	Y	
\tabH@driver	184,	\y	331, 335, 343,
	191, 443, 457, 458, 460, 470, 473		373, 407, 491, 493, 494, 496, 499
\tabH@driver@dvi	452	Z	
\tabH@headerps	454, 510	\z@	382, 385, 386, 387, 388, 404
\tabH@interrowfill	159, 161	\zap@space	99
\tabH@interrow	151, 153		
\tabH@interrowstart	175, 177		