

# tabu

tabu and longtabu

Flexible L<sup>A</sup>T<sub>E</sub>X tabulars

FC

2011/01/25 – version 2.0

1	2	3
5	6	7
8	9	4

$\eta$	$\theta$	$\beta$
$\phi$	$\epsilon$	$\alpha$
$\lambda$	$\gamma$	$\kappa$

## Abstract

This package defines a single environment `tabu` to make all kinds of tabulars in text or in math mode provided that they do not split across pages.

Starting from version 1.5, the environment `longtabu` is provided to make tabulars that can stretch out on several pages. `longtabu` works like `tabu` and is defined only if the package `longtable` is loaded.

`tabu` is more flexible than `tabular`, `tabular*`, `tabularx` and `array` and extends the possibilities. All tabulars in this document were made with the `tabu` environment. Widths computations for `tabu`  $X$  columns is *optimised* compared to `tabularx` implementation. Last (but not least) `tabu` is fully compatible with any package that provides commands to format tabulars. Indeed, `tabu` does not modify any of the macros of `array.sty`.

`tabu` requires  $\epsilon$ -T<sub>E</sub>X and the standard package `array.sty`. Natural widths of columns are computed by (but not printed by) the code of `varwidth` by D. Arseneau (for the cases of  $X$  columns in `tabu spread` and *negativ width coefficients*<sup>1</sup>). Finally `longtabu` is based on `longtable`.

## Contents

Summary of the features provided by <code>tabu</code> .....	3
1 The <code>tabu</code> environment .....	4
1.1 <code>tabu</code> , <code>tabu to</code> and <code>tabu spread</code> .....	4
1.2 <code>longtabu</code> , <code>longtabu to</code> and <code>longtabu spread</code> .....	4
1.3 <code>tabu X</code> columns .....	5
1.3.1 $X$ columns with “ <code>tabu spread</code> ” .....	
1.3.2 <i>Negativ width coefficients for <math>X</math> columns</i> .....	
1.3.3 <i>Multicolumn in <code>tabu</code></i> .....	
1.3.4 <i>Nesting <code>tabus</code> with <math>X</math> columns</i> .....	
1.4 Lines inside <code>tabu</code> .....	8
1.4.1 <i>First important remarks</i> .....	
1.4.2 <i>Vertical lines: <code> </code> has an optional parameter</i> .....	
1.4.3 <i>More style for horizontal lines: <code>\tabucline</code></i> .....	
1.5 <code>tabu</code> in math mode .....	10
1.6 Verbatim inside <code>tabu</code> with $X$ columns .....	10
1.7 Printing numbers inside <code>tabu</code> with <code>numprint</code> and <code>siunitx</code> .....	11
1.7.1 <i>Just make it easy !</i> .....	
1.7.2 <i>You should know how it works...</i> .....	
1.8 Paragraph indentation .....	12
1.9 <code>delarray</code> shortcuts .....	12
2 Modifying the font and the alignment in one row: <code>\rowfont</code> .....	12
3 Controlling vertical spacing of lines: <code>\extrarowsep</code> and <code>\tabulinesep</code> .....	13
3.1 <code>\extrarowheight</code> <code>\extrarowdepth</code> and <code>\extrarowsep</code> .....	13
3.2 <code>\abovetabulinesep</code> <code>\belowtabulinesep</code> and <code>\tabulinesep</code> .....	13
4 Saving and restoring a <code>tabu</code> : <code>\savetabu</code> , <code>\usetabu</code> and <code>\preamble</code> .....	14
5 Differences between <code>tabu</code> , <code>tabular</code> , <code>tabularx</code> and <code>longtable</code> .....	16
5.1 Paragraph indentation .....	16
5.2 Custom environments .....	16

This documentation is produced with the DocStrip utility, and required `tabu` with its `linegoal` option.

- To get the package, run: `etex tabu.dtx`
- To get the documentation run (thrice): `pdflatex tabu.dtx`
- To get the index, run: `makeindex -s gind.ist tabu.idx`

The `.dtx` file is embedded into this pdf file thank to `embedfile` by H. Oberdiek.

1. The documentation is compiled with `enumitem-zref`

5.3	Inversion of tokens .....	16
5.4	Improved process for rewriting columns ( <i>for keen readers</i> ) .....	16
<b>6</b>	<b>The package options .....</b>	<b>18</b>
6.1	The <code>debugshow</code> package option .....	18
6.2	The <code>linegoal</code> package option .....	18
6.3	The <code>delarray</code> package option .....	18
6.4	The <code>light</code> package option .....	18
<b>7</b>	<b>Corrections of some bugs (<i>available only inside tabu</i>) .....</b>	<b>19</b>
7.1	Correction for <code>colortbl</code> and <code>arydshln</code> : compatibility with <code>delarray</code> .....	19
7.2	Correction for <code>arydshln</code> : @ columns .....	19
<b>8</b>	<b>To do for even better tabus .....</b>	<b>19</b>
<b>9</b>	<b>IMPLEMENTATION .....</b>	<b>20</b>
9.1	Identification, requirements and options .....	20
9.2	Some constants .....	21
9.3	<code>\tabu</code> , <code>\endtabu</code> , <code>\longtabu</code> and <code>\endlontabu</code> .....	26
9.4	Footnotes inside <code>tabu</code> .....	29
9.5	Flow chart of expansion .....	31
9.5.1	<i>General case: tabu, tabu to and tabu spread</i>	
9.5.2	<i>tabu to with X column</i>	
9.5.3	<i>tabu spread with X column</i>	
9.6	<code>tabu X</code> column definition .....	33
9.7	Before trials to reach the target: setup after <code>\@mkpream</code> .....	35
9.8	One trial after another .....	36
9.9	Algorithms: Measuring the <code>tabu</code> .....	38
9.9.1	<i>The arithmetic of X columns</i>	
9.9.2	<i>The arithmetic of X columns for tabu spread</i>	
9.9.3	<i>Reporting in the .log file (debugshow option)</i>	
9.9.4	<i>Measuring the natural width for tabu spread and negativ width-coefficients</i>	
9.10	<code>\everycr</code> setup and <code>\tabupantomline</code> .....	42
9.11	Lines inside <code>tabu</code> .....	43
9.11.1	<i>Vertical lines</i>	
9.11.2	<i>Horizontal lines: \tabucline</i>	
9.12	Verbatim inside <code>tabu</code> with <code>X</code> columns .....	47
9.13	Numbers in <code>tabu</code> .....	47
9.13.1	<i>\tabudecimal</i>	
9.14	<code>\extrarowsep</code> and <code>\tabulinesep</code> .....	48
9.15	<code>\savetabu</code> .....	50
9.16	<code>\rowfont</code> .....	51
9.16.1	<i>Setting font and alignment specification</i>	
9.16.2	<i>Preparing stuff to be able to use \rowfont</i>	
9.16.3	<i>Neutralisation of glues and alignment modification</i>	
9.17	Utilities .....	56
9.17.1	<i>tabu \fbox</i>	
9.17.2	<i>tabu \fcolorbox</i>	
9.17.3	<i>\centering, \raggedright, \raggedleft</i>	
9.18	Corrections .....	57
9.18.1	<i>delarray comptability fix for colortbl and arydshln</i>	
9.18.2	<i>arydshln @ columns</i>	
9.18.3	<i>arydshln, colors without colortbl and empty p columns</i>	
<b>10</b>	<b>References .....</b>	<b>60</b>
<b>11</b>	<b>History .....</b>	<b>60</b>
	[2011/01/25 v2.0] .....	60
	[2011/01/15 v1.9] .....	60
	[2010/12/28 v1.8] .....	60
	[2010/12/18 v1.7] .....	61
	[2010/12/07 v1.5] .....	61
	[2010/11/22 v1.4] .....	61
	[2010/11/18 v1.3] .....	61
	[2010/11/15 v1.2] .....	62
	[2010/10/28 v1.1] .....	62
<b>12</b>	<b>Index .....</b>	<b>62</b>

## Summary of the features provided by **tabu**

<b>tabu</b>	is like <b>tabular</b> in text mode and like <b>array</b> in <b>math mode</b> when there is no <b>X</b> column in its preamble
<b>longtabu</b>	is like <b>longtable</b> with the possibility to use <b>tabu</b> <b>X</b> columns and vertical lines with the extended syntax.
<b> [width,color]</b>	vertical lines have an optional parameter.
<b>{tabu} to &lt;dimen&gt;</b>	specifies the target width of the whole tabular. This is like <b>tabular*</b> with an automatic stretchability that can be overwritten with <code>@{\extracolsep{dimen}}</code> in front of the preamble.
<b>{tabu} spread &lt;dimen&gt;</b>	has no equivalent in L <sup>A</sup> T <sub>E</sub> X: the final width is <i>&lt;dimen&gt;</i> wider than the natural width that can be obtained with <b>spread Opt.</b>
<b>X[coef,align,type]</b> <b>X[coef,align,type,\$]</b>	X columns widths are adjusted in order for the whole tabular to fit the target width. The target width is a dimension either: <ul style="list-style-type: none"> <li>→ directly specified with <b>{tabu} to&lt;dimen&gt;</b></li> <li>→ computed from the natural width: <b>{tabu} spread&lt;dimen&gt;</b></li> <li>→ by default <code>\linewidth</code> (or <code>\linegoal</code> with the <b>linegoal</b> package option).</li> </ul> <b>coef</b> scales the widths of the <b>X</b> columns, if there are more than one <b>X</b> column. <b>align</b> is either <b>r</b> , <b>c</b> , <b>l</b> or <b>j</b> (or <b>R C L J</b> ) and <b>type</b> can be <b>p</b> (default), <b>m</b> or <b>b</b> . <b>X[\$]</b> makes a math <b>X</b> column ( <i>ie.</i> <code>&gt;{\$}X&lt;{\$}</code> )
<b>X[-coef,align,type]</b>	X columns widths are first computed with the absolute value: <code> coef </code> . Then the width is made narrower down to the natural width of the column if possible. In any case, the final width does not exceed the one obtained with <b>X[ coef ]</b> . The final <b>tabu</b> can be narrower than its <b>target</b> if: <ul style="list-style-type: none"> <li>• all <b>coefs</b> are negativ, and</li> <li>• no <b>target</b> has been specified (<i>ie.</i> the default <b>target</b> is used).</li> </ul> In this case, <b>tabu</b> behaves as if it was said: <b>{tabu} spread Opt.</b>
<b>\rowfont[align]{font spec}</b>	Modify the font and optionally the alignment of each cell in one row.
<b>\tabulinesep=&lt;dimen&gt;</b>	More control on vertical spacing of lines in a way very close to <b>cellspace</b> .
<b>\extrarowsep=&lt;dimen&gt;</b>	In addition to <code>\extrarowheight</code> , <b>tabu</b> provides <code>\extrarowdepth</code> . But <b>\tabulinesep</b> generally gives better results.
<b>\savetabu{user-name}</b>	Saves the <b>tabu</b> preamble and its parameters. The command must appear at the end of a line: after <code>\</code> or <code>\hline</code> .
<b>\usetabu{user-name}</b>	Makes a <b>tabu</b> of exactly the same shape as the one saved with <b>\savetabu</b> . All parameters ( <b>target</b> , <b>preamble</b> , <b>stretch</b> <i>etc.</i> ) are restored. This command is put alone in the preamble in place of the columns specifications.
<b>\preamble{user-name}</b>	Makes a <b>tabu</b> with the same preamble as the one saved with <b>\savetabu</b> . The <b>preamble</b> only is restored, not the <b>target</b> nor any other parameter. This command is put alone in the preamble in place of the columns specifications.
<b>\tabuphantomline</b>	inserts a phantom ( <i>ie.</i> invisible) line inside the <b>tabu</b> May be usefull with <b>\multicolumn</b> in some cases.
<b>\tabucline[spec]{start-stop}</b>	Draws a line comparable to <code>\hline</code> . The line <i>&lt;spec&gt;</i> can contain information for making a dash or dotted line (f.ex. <code>[on 3pt off 6pt]</code> ) and a color name. The line spec can also be defined with <b>\tabulinestyle</b>
<b>\tabulinestyle{name=spec,...}</b>	Defines a line style for use with <b>\tabucline[name]</b> or with <b> [name]</b>
<b>\tracingtabu</b> <b>\tracingtabu=2</b>	Reports informations in the .log file about the steps of the algorithm for <b>tabu</b> <b>X</b> columns, and the informations saved by <b>\savetabu</b> .

| **X** **\usetabu** and **\preamble** are defined as new column types loaded only inside the **\@mkpream** group inside the **tabu** environment.

# 1 The tabu environment

## 1.1 tabu, tabu to and tabu spread

```
\begin{tabu} [pos] {tabular preamble}
\begin{tabu} to <dimen> [pos] {tabular preamble}
\begin{tabu} spread <dimen> [pos] {tabular preamble}
```

The `tabu` environment behaves exactly like `tabular`: the preamble is parsed by the macros in `array.sty` with no modification. `tabu` improves `tabular` and `array`:

- **footnotes** and index words are allowed inside `tabu`, unlike `tabularx`, footnote links are not broken when compiled with `hyperref`. The syntax `\footnote[<number>]{<text>}` is allowed in `tabu` and `longtabu` (this is not implemented for `longtable` yet...)
- **X columns** are implemented with an *optional* parameter for the **width-coefficient** (which can be *negativ*: see next section), the **alignment** (`r`, `c`, `l`, or `j`, and `R`, `C`, `L` or `J` for `ragged2e` settings) and the **column type** (`p`, `m`, or `b`).  
`tabu` has a *default target width* when used with **X columns**, making nesting even easier.
- You are used to the `tabular` environment in text mode, and `array` in math mode, but `tabu` works in both modes and its name does not change... **X columns** are also possible in math mode; `delarray` shortcuts for delimiters are available in both math and text modes.
- A `tabu` environment can contain another tabular of any kind: `tabular`, `tabular*`, `tabularx` or `tabu` itself can be placed in any cell of a `tabu`. Conversely, `tabu` can be placed in a `tabular`, `tabularx` *etc.*.
- `tabu` provides facilities for *vertical and horizontal lines*, and for the insertion of *verbatim text* inside **X columns**.
- `tabu` is more than compatible with `arydshln` (for dashed and dotted lines) and `colortbl`: actually some corrections of those packages are loaded as soon as you enter a `tabu` environment. Compatibility with `delarray`, `hhline`, `makecell`, `booktabs`, `siunitx`, `dcolumn`, `warpcol`, *etc.* is fine too. When you are inside a `tabu` environment, you can use `\raggedleft`, `\raggedright` and `\centering` without special care about `\arraybackslash` and conversely `\\` has its “normal” meaning inside a list of items that may appear in a **X column**...

`\begin{tabu} to<dimen>` is like `tabular*` but the inter-columns space is given a stretchability of `1fil`, in other words `@{\extracolsep{0pt plus 1fil}}` is inserted by default at the beginning of the tabular preamble, unless another value for `\extracolsep` is specified. Therefore “`tabu to`” fills in width the specified `<dimen>`.

`\begin{tabu} spread<dimen>` does a tabular whose width is `<dimen>` wider than its natural width. `@{\extracolsep{0pt plus 1fil}}` is inserted by default if `<dimen> > 0`.

## 1.2 longtabu, longtabu to and longtabu spread

```
\begin{longtabu} [l | c | r] {tabular preamble}
\begin{longtabu} to <dimen> [l | c | r] {tabular preamble}
\begin{longtabu} spread <dimen> [l | c | r] {tabular preamble}
```

`longtabu` is just like `tabu` but page breaks are allowed between rows of the table. `longtabu` is based on the `longtable` package which must be loaded, and all features of the `longtable` environment works inside `longtabu`: `\endhead`, `\endfirsthead`, `\endfoot`, `\endlastfoot` and `\caption`.

`longtabu` enhances the possibilities of `longtable` with the possibility to use **X columns** and line specification for *vertical rules*. `longtabu` is thus much easier than `ltxable`.

The following commands provided for `tabu` do not work with `longtabu`:

tabu command	Not available	Not implemented	Comment
<code>\tabucline</code>		✱	<code>\tabucline</code> does not care of page breaks presently: use <code>\hline</code> instead.
<code>\usetabu</code>	✗		but <code>\savetabu</code> and <code>\preamble</code> work.
mathematical mode	✗		<code>longtable</code> is not designed to work in math mode.
delarray shortcuts	✗		a delimiter cannot be spanned over pages...
<code>\tabuphantomline</code>	✗		useless inside <code>longtabu</code>

However, `tabu` X columns, `\rowfont`, `\verbatim` and `\tabudecimal` work inside `longtabu`.

1.3 tabu X columns

`tabu` X columns can be viewed as an enhancement of `tabularx` X columns, but do not interact with them, for they are defined only for a short time during the parsing of the preamble:

- width coefficients** can optionally be given to X columns  
ex. `X[2.5]X[1]` is the same as `X[2.5]X` and the same as `X[5]X[2]`  
This means that the first X column will be two and a half wider than the second one or that the first X column width will be 5/7 of the whole tabular width.

X[2.5]	X
--------	---

- negativ width coefficients** can be given to X columns:  
ex. `X[-2.5]X[1]` or `X[-2.5]X` or `X[-5]X[2]`  
In this case, the first X column will be *at most* two and a half wider than the second one, and if the *natural width* of the first X column is finally less than  $2.5 \times$  (the width of the second column) then it will be narrowed down to this natural width.

The following tabus have the same preamble:

`\begin{tabu}` to `\linewidth{ | X[-2.5c] | X[c] | }`:

X[-2.5]	X
Negativ coefficients make X columns close to standard l, c and r columns.	X

- horizontal alignment specification is made easier with `X[5,r]X[2,c]` for example. Vertical alignment can be specified as well with `X[5,r,m]X[2,p,c]` (commas are not required, but `X[2cm]` or `X[4pc]` could be misunderstood – not by T<sub>E</sub>X: by you...).

Modifier	Meaning	Default
l, c, r, j, L, C, R, J	left, centered, right, justified	j
p, m, b	X column is converted into p, m or b column	p
\$	X[\$] is a shortcut for: <code>&gt;{\$}X&lt;{\$}</code>	

- `tabu` X columns can be spanned with `\multicolumn`.
- `tabu` X columns can be used with “`tabu spread`” for small tabulars.
- `tabu` X columns can contain any type of `tabular`, `tabular*`, `tabularx` or `tabu` without special care about the syntax. `tabu` can also be put inside `tabular`, `tabular*` and `tabularx`. As long as `tabu` with X columns has a *default target*, nesting `tabu` with X columns is easy. Furthermore, the default global alignment of a nested `tabu` is t (for top) while the default global alignment of a `tabu` in a paragraph is c (for centered).
- The “algorithm” (or the arithmetic) to get the target width for `tabu` X columns is the same as the one used by `tabularx`. `\hfuzz` is the “tolerance” for the whole tabular width. We use  $\varepsilon$ -T<sub>E</sub>X `\dimexpr` instead of T<sub>E</sub>X primitives (with round/truncate bias correction).
- Convergence to the target width is optimised: the `\halign` preamble is not re-built at each trial, but only expanded again, until the target is reached. Though optimized, the process is the same as the one implemented for `tabularx` and in particular the content of

the `tabu` environment is collected as soon as a `tabu X` column is found in the preamble. This implies restrictions on catcode modifications and verbatim text inside a `tabu` with `X` columns.

- If the width of the whole tabular is not specified with “`tabu to`” it is considered to be `\linewidth`. The `linegoal` package option makes the default width equal to `\linegoal`. Compilation must then be done with pdf<sub>T</sub>E<sub>X</sub> either in `pdf` or `dvi` mode, and package `linegoal` is loaded. `\linegoal` requires pdf<sub>T</sub>E<sub>X</sub> for its `\pdfsavepos` primitive and the `zref-savepos`: if the `tabu` is not alone in its paragraph *ie.* if the target is not `\linewidth`, then two compilations (or more) are required to get the correct target. Default target for nested `tabu` environments is always `\linewidth`, which equals to the column width inside `p`, `m`, `b` and `X` columns.
- As long as the `\halign` content is expanded more than once, protections against counters incrementation, *whatsits* (*write*) index entries, footnotes *etc.* are set up: the mechanism of `tabularx` is reimplemented and enhanced for `tabu X` columns. `\tabuDisableCommands` can be used to neutralize the expansion of additional macros during the trials.

### X columns with “`tabu spread`”

`tabu X` columns can be used with “`tabu spread`” to adjust the column widths of tabulars that contain only small pieces of text. The question is: how to make a tabular the width of the line, with 6 columns; the columns 1, 2, 5 and 6 are of equal widths and the widths of columns 3 and 4 are only one half. As possible solution:

```
\begin{tabu} to\linewidth{ |X[2]|X[2]|X|X|X[2]|X[2]| } \hline
1 & 2 & 3 & 4 & 5 & 6 \\ \hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

But the text in each cell is very short: one single character, and you prefer the table to be tight, but don’t know the exact width of the whole:

```
\begin{tabu} spread 0pt{ |X[2]|X[2]|X|X|X[2]|X[2]| } \hline
1 & 2 & 3 & 4 & 5 & 6 \\ \hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

But now it’s definitely too narrow, then give it some more space:

```
\begin{tabu} spread 2in{ |X[2]|X[2]|X|X|X[2]|X[2]| } \hline
1 & 2 & 3 & 4 & 5 & 6 \\ \hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

`tabu spread` is useless with long columns: the following tabular was made with this preamble:

```
\begin{tabu} spread 3cm{@{}X[9]X[4]|X|}
```

"Like the air we breathe, Sherlock Holmes is everywhere. His pipe-smoking, deer stalked image peers at us from ads in Yellow Pages, to signs for neighbourhood crime-watch; from billboards to the classroom; from film and television to the public library, and now over the Internet. He long ago transcended the boundaries of 19th Century London <sup>2</sup> to become an international best-seller and has been accepted as part of British folklore. Holmes is alive to millions."	There the text was too long, and <code>tabu spread</code> behaves as if you didn’t give it a target.  The result of this example is the same as if one had written <code>\begin{tabu}to\linewidth.</code>	Sherlock Holmes
The “official” web site: <a href="http://www.sherlockholmes.com/">http://www.sherlockholmes.com/</a>		

In the preamble, `@{}` means that the margin is removed.

2. Capital of the U.K. (too see a linked footnote)



## Negativ width coefficients for X columns

```
\begin{tabu}{|[3pt ForestGreen]X[-1m]|X[c m]||[3pt ForestGreen]}
\tabucline[3pt ForestGreen]-
$\begin{tabu}({X[-1$]X[-1$c]})$ % equivalent to spread 0pt
\alpha & \beta \\
\gamma & \delta + \epsilon + \zeta + \eta + \theta
\end{tabu}$
&
This is a tabu with negativ width coefficients for \texttt X columns
\\ \tabucline[3pt ForestGreen]-
\end{tabu}
```

$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta + \epsilon + \zeta + \eta + \theta \end{pmatrix}$	This is a tabu with negativ width coefficients for X columns
--	--

$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta + \epsilon + \zeta + \eta + \theta \end{pmatrix}$	And this is the same with <code>\tabulinesep</code> set to 2pt.
--	---

## Multicolumn in tabu

### `\tabuphantomline`

The process of `\multicolumn` implies the T<sub>E</sub>X primitive `\omit` which discard the tabular preamble for the spanned columns. Discarding the preamble means discarding the information about the widths of the columns. This explains why the following example does not work properly:

```
\begin{tabu}{|X|X|X[2]|} \tabucline-
\multicolumn2{|c|}{Hello} & World \\ \tabucline-
\end{tabu}
```

Hello	World
-------	-------

The correct result can be obtained by the mean of a phantom line, that will remain invisible unless your preamble contains special `@` or `!` columns that prints some text:

```
\begin{tabu}{|X|X|X[2]|} \tabucline-
\multicolumn2{|c|}{Hello} & World \\ \tabucline-
\tabuphantomline
\end{tabu}
```

Hello	World
-------	-------

Remember you may need `\tabuphantomline` in conjunction with `\savetabu` and `\usetabu` with `\multicolumn`. Even if it is possible to add a `\tabuphantomline` in any line of the `tabu`, it is a good practice to append it *at the end* of the `tabu`, for it may introduce undesirable side effects on vertical alignment otherwise, when `tabu` is nested inside another tabular.

## Nesting tabus with X columns

This section should contain some examples but I've no time presently...

However, this document has plenty of nested `tabus`!

## 1.4 Lines inside tabu

### First important remarks

The features provided in this section are quite experimental: they are not generally taken for good typography. You can use `tabu` with package `booktabs` for example, which provides properly designed commands for horizontal rules in tabulars. `arydshln` is pretty good too, but it modifies a huge amount of macros of `array.sty`, something that `tabu` does not.




Lines in `tabu` printed in this document are mostly made with `booktabs`.

### Vertical lines: `|` has an optional parameter




Inside `tabu` environment, the vertical line marker `|` has an *optional* argument which is the width of the vertical rule. The default width remains `\arrayrulewidth` of course. The optional argument for `|` can also contain the name of a color. *color names* are only possible, not a color specification by the mean of a color model. The width of the line if specified, must come before the color name and... as for `X` columns parameters, commas are optional.

Example:

---

	Hello		World	
The <code>tabu</code> you see on left <code>\begin {tabu}{ [5pt] c c [5pt] }</code> was made with the code <code>Hello &amp; World</code> displayed on the right. <code>\end {tabu}</code>				

---

	Hello		World	
The <code>tabu</code> you see on left <code>\begin {tabu}{ [5pt,red] c c [5pt blue] }</code> was made with the code <code>Hello &amp; World</code> displayed on the right. <code>\end {tabu}</code>				

---

This example was printed inside a `tabu` whose preamble is: `X[-1m]X[m]X[2m]`.

Note that it is always a good idea to protect the optional argument with braces: `[{...}]`. But it's not necessary because `tabu` takes care the `|` token to be rewritten before any other column type (just after `*` however, for obvious reasons). But if you use the optional argument for the vertical line into a user-defined column type (declared with `\newcolumnntype` for example), you can get an error. In this case, it is compulsory to protect the optional argument by braces. Finally, it's not very often that a user-column type contains a vertical bar...

### More style for horizontal lines: `\tabucline`

`\tabucline[<style or spec.>]{start-end}`

`\tabucline` is an attempt to give a versatile command to make horizontal lines:

- `\tabucline` is pretty good with vertical lines even if the thickness of the line grows up,
- `\tabucline` takes care of `\extrarowheight`,
- `\tabucline` can make horizontal dashed lines, with a `pgf/TikZ` syntax:  
`\tabucline[<width> on<dash> off<gap>]{<first column>-<last column>}`
- alternatively, you can give `\tabucline` a `\hbox` to make a leader with it: The *<spec.>* must then begin with `\hbox`, `\box` or `\copy`,
- finally you can give `\tabucline` a color *name*, after the line specification.

Any parameter can be omitted.



[1pt on 1.5pt off 2pt]  
  
[1.5pt]  
  
default  
  
[on 2pt red]

<code>\tabucline[1pt on 1.5pt off 2pt]{1-4}</code>	draws a horizontal dashed line of width 1pt. Dashes are 1.5pt long and gap width is 2pt. The line is drawn between columns 1 and 4. Here there are only 2 columns and the line stops at column 2.
<code>\tabucline[1.5pt]{-}</code>	draws a horizontal solid line of width 1.5pt between the first and the last column.
<code>\tabucline{2-}</code>	draws a horizontal solid line of width <code>\arrayrulewidth</code> between the second column and the last one.
<code>\tabucline[on 2pt red]{-5}</code>	draws a horizontal dashed line between columns 1 and 5 of width <code>\arrayrulewidth</code> . Dashed are 2pt long and gap width is 4pt (the default).

`\tabucline*{<style or spec.>}{start-end}`

Starting from [version 2.0](#) use preferably the vertical spacing adjustment parameters provided by `\tabulinesep` and `\extrarowsep`.

For fine tuning, the star form `\tabucline*` can be used to keep the vertical lines that might cross the horizontal line. As a consequence, the content of special @ of ! columns will interrupt the horizontal line either. This might be usefull when `\extrarowheight` is high. Example:

Hello	world
<hr/>	
Hello	world

`\tabucline`

Hello	world
<hr/>	
Hello	world

`\tabucline*`

Hello	world
<hr/>	
Hello	world

`\firsthline` and `\hline` and `\lasthline`

In each table, `\extrarowheight` is equal to 8 pt.








`\tabuclines` tries to put the line in the middle, so that the text is centered in its cell. `\tabuclines*` tries to take care of vertical lines. Well this is not perfect: it works pretty well with simple vertical lines, but not really with double lines, triple lines with colours etc... But it can help in simple cases...

`\tabulinestyle{<style=spec., style=spec., style=spec. ...>}`

This command defines a line style to be used in the first optional argument of `\tabucline`:

Define the line style	Use the line style
<code>\tabulinestyle{myline=0.4pt on 1.5pt off 1pt red}</code>	<code>\tabucline[myline]{-}</code>
<code>\tabulinestyle{myleader=\hbox{\$\scriptstyle\star\$}}</code>	<code>\tabucline[myleader]{1-3}</code>

This way it is easy to change dashed lines into solid ones

Dashed or dotted	Dash      Gap	Dash   Gap...   ...or leader
And below is the default	 This one was thick	     
		Interesting ?

Now we stop  
it because things  
are be- com-  
ing too ugly  
and go to have  
a look at `tabu`  
in mathematical  
mode...

### 1.5 tabu in math mode

On the left, you can see the famous Maxwell-Lorentz equations for electromagnetic field in vacuum, published in 1873.

$$\left\{\begin{array}{l} \operatorname{div} \vec{E} = \frac{\rho}{\epsilon_0} \\ \operatorname{div} \vec{B} = 0 \\ \overrightarrow{\operatorname{rot}} \vec{E} = -\frac{\partial \vec{B}}{\partial t} \\ \overrightarrow{\operatorname{rot}} \vec{B} = \mu_0 \vec{j} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \end{array}\right.$$

In this example, the big `tabu` is: `\begin{tabu} to\linewidth{X[1.5]X[l$]}`.

The nested `tabu` (in math mode) uses `delarray` shortcut: its preamble is:

`\begin{tabu}({rl})`.

`\tabulinesep` has been set to **3pt**.

Horizontal rules are `booktabs` `\toprule`, `\midrule` and `\bottomrule`.

array	tabu	tabu spread 1em
$\left  \begin{array}{cc} \alpha & \beta \\ \gamma & \delta \end{array} \right $	$\left  \begin{array}{cc} \alpha & \beta \\ \gamma & \delta \end{array} \right $	$\left  \begin{array}{cc} \alpha & \beta \\ \gamma & \delta \end{array} \right $

Here, vertical lines are made with `delarray` shortcuts: `$$\begin{tabu} spread 1em |{cc}|`

Vertical lines inside the tabular preamble gives:

$$\left| \begin{array}{cc} \alpha & \beta \\ \gamma & \delta \end{array} \right|$$

This was an example of `\savetabu... \usetabu` to keep the alignment.

### 1.6 Verbatim inside tabu with X columns

The process of computing `X` columns widths implies to make “trials”: this means that the tabular is first printed in internal mode, inside a box which is measured until the target width is reached. Such trials require to collect the content of the `tabu` environment. Therefore, the category codes of the characters read in the environment are fixed, and `\verb` commands are not allowed.

`\verbatim{<general text>}`

To get round this limitation, `tabu` provides the command `\verbatim` which allows to put some pieces of verbatim text inside a `tabu`: it is based on  $\epsilon$ -T<sub>E</sub>X `\scantokens` primitive and is defined only inside the `tabu` environment. `\verbatim` has some other limitations you must know:

<code>^^@</code>	can be used to get a carriage return inside <code>\verbatim</code>
<code>^</code>	consequently <code>^</code> is a normal subscript character
<code>{ }</code>	curly braces must be equilibrated
<code>%</code>	The percent character <code>%</code> is for commentaries: the content of the environment is collected before <code>\verbatim</code> does its job, therefore commentaries are removed...

If you need more verbatim inside a `tabu` environment:

- you should avoid the use of `X` columns
- or if you really want to use `X` columns, you must save your verbatim text before the `tabu`, for example with the `SaveVerbatim` environment provided by the package `fancyvrb`.

## 1.7 Printing numbers inside tabu with numprint and siunitx

Just make it easy !

`tabu` provides a *facility* to print numbers inside columns. This facility is not implemented to replace `siunitx` `S` and `s` columns or `numprint` `n` and `N` columns or other packages that provide alignment such as `warpcol`, `dcolumn` or `rccol`. It just make easy to apply a macro you get already on each number in a column of a `tabu`.

`\tabudecimal` has been developped mainly because it makes possible to align numbers inside `tabu` `X` columns.

`\tabudecimal{⟨user-macro⟩}`

`\tabudecimal` can be used in the preamble of a `tabu` before a column specification. The `⟨user-macro⟩` is a macro with one parameter that has to be defined before.

Example with `\numprint`:

```
\def\usermacro#1{\numprint[\officialeuro]{\zap@space #1 \@empty}}
\nprounddigits{2} \npprintnull \npthousandsep{\,} \npunitseparator{~}

\begin{tabu}{|[GreenYellow]*2>{\tabudecimal \usermacro }X[r]| [GreenYellow]}
  \rowfont [c]{\bf } January & February \\\
12.324 & 745.32 \\\
21.13 & 0 \\\
213.3245 & 12.342 \\\
2143.12 & 324.325 \\\
\end{tabu}
```

January	February	...
12,32 €	745,32 €	...
21,13 €	0,00 €	...
213,32 €	12,34 €	...
2 143,12 €	324,33 €	...

Example with `\SI`:

```
\def\usermacro#1{\SI[group-four-digits=true,           % thousand separator
                      round-mode=places,                 % round numbers
                      round-precision=2,                  % with 2 decimal digits
                      round-integer-to-decimal=true,      % add trailing 0 if necessary
                      per-mode=symbol]{#1}{\officialeuro\per\kilo\gram}}

\begin{tabu}spread Opt{|[GreenYellow]*2>{\tabudecimal \usermacro }X[r]| [GreenYellow]} ...
```

January	February	...
12.32 €/kg	745.32 €/kg	...
21.13 €/kg	0.00 €/kg	...
213.32 €/kg	12.34 €/kg	...
2 143.12 €/kg	324.33 €/kg	...

As you can see, the columns widths are exactly the same, whatever their content.

Here `\tabulinesep` has been set to *3pt*.

You should know how it works...

Yes you should know how it works to avoid problems. `tabu` has a small scanner based on `\futurelet` to grab all numbers, blank spaces, commas and dots + and – sign and also the letter `e` and `E` for exponents. The scanner stops as soon as something else than a number, blank space, comma, dot, +, –, `e`, `E` is found, and even if it is a macro that contains a number.

This explains why there is `\zap@space` in the definition of `\usermacro`: because the scanner scans blank spaces and because `\numprint` does not allow blank spaces in its mandatory argument, quite strangely...

## 1.8 Paragraph indentation

`tabu` takes care of paragraph indentation when it is used with `X` columns and its default target, no matter if it has been loaded or not with the `linegoal` option. Example with L<sup>A</sup>T<sub>E</sub>X default: `\parindent = 20pt`.

This is `tabu` with its default target in an indented paragraph.

This is `tabu` with its default target, preceded by `\noindent`

This is `tabularx` with target: `\linewidth` in an indented paragraph.

This is `tabularx` with target: `\linewidth`, preceded by `\noindent`

## 1.9 delarray shortcuts

When you enclose your tabular with math delimiters using `delarray` shortcuts, `tabu` tries to reach its target for the whole: the tabular and the delimiter(s). You can see the difference:

with overfull hboxes  
(17.5pt and 17.8pt two wide)  
for tabularx

$\left( \begin{array}{l} \text{This is tabu with delarray} \\ \text{shortcuts for paren-} \\ \text{thesis around.} \end{array} \right)$	$\left\{ \begin{array}{l} \text{This is tabu with de-} \\ \text{larray shortcuts for curly} \\ \text{brackets around.} \end{array} \right\}$
$\left( \begin{array}{l} \text{This is tabularx with de-} \\ \text{larray shortcuts for paren-} \\ \text{thesis around.} \end{array} \right)$	$\left\{ \begin{array}{l} \text{This is tabularx with de-} \\ \text{larray shortcuts for curly} \\ \text{brackets around.} \end{array} \right\}$

## 2 Modifying the font and the alignment in one row: `\rowfont`

$$\backslash\mathrm{rowfont}[\langle alignment \rangle]\{\mathrm{font\ specification}\}$$

Inside a `tabu` environment, you can modify the font for each cell in a row. `\rowfont` has priority over column font specification, exactly like `\rowcolor` (package `colortbl`) has priority over `\columncolor`.

The alignment of each cell in one row can also be changed to:

l = left	or for ragged2e settings:	L
c = center		C
r = right		R
j = justify		J

Any other value for the optional `<alignment>` parameter is silently ignored. If `ragged2e` is not loaded, `L` `R` `C` and `J` are synonymous with the lowercase equivalent.

```
\begin{tabu}{|X|X[-1]|} \tabucline-
\rowfont[c]\bfseries
This &Is \\ \tabucline[on 2pt,blue]-
\xpackage{tabu} & \xpackage \tabu \tabucline[off 2pt blue]-
\rowfont[r]\itshape
for & \texttt{tabu} and \texttt{longtabu} \\ \tabucline-
\end{tabu}
```

<b>This</b>	<b>Is</b>
tabu	package
<i>for</i>	<i>tabu and longtabu</i>

### 3 Controlling vertical spacing of lines: `\extrarowsep` and `\tabulinesep`

#### 3.1 `\extrarowheight` `\extrarowdepth` and `\extrarowsep`

$$\begin{aligned}\backslash extrarowheight &= \langle \textit{dimen} \rangle \\ \backslash extrarowdepth &= \langle \textit{dimen} \rangle\end{aligned}$$

`\extrarowheight` is a dimension defined in `array.sty` whose purpose is to add – inconditionally – some more height to each line in a tabular. Similarly, `\extrarowdepth` (defined in `tabu.sty`) is added – inconditionally as well – to the depth of each line.

Both `\extrarowheight` and `\extrarowdepth` are scaled by `\arraystretch`<sup>3</sup> if `\arraystretch > 1`.

$$\backslash extrarowsep = \langle \textit{dimen} \rangle$$

`\extrarowsep` makes the assignment for both `\extrarowheight` and `\extrarowdepth`. The macro can be prefixed by `\global`<sup>4</sup>.

Examples (with `\tabustrutrule = 2pt` to see the struts):

Standard c column	X[-1,c] columns	Math mode	Mixed: X[-1c]X[-1c\$]
One Two	One Two	$\alpha \quad \beta$	First $\frac{\partial \Phi}{\partial \theta}$
Three Four	Three Four	$\frac{\Phi}{\theta} \quad \Gamma_x^t$	Second $\frac{d\theta}{dt}$

$\backslash extrarowsep = 3mm$

Standard c column	X[-1,c] columns	Math mode	Mixed: X[-1c]X[-1c\$]
One Two	One Two	$\alpha \quad \beta$	First $\frac{\partial \Phi}{\partial \theta}$
Three Four	Three Four	$\frac{\Phi}{\theta} \quad \Gamma_x^t$	Second $\frac{d\theta}{dt}$

$\backslash extrarowsep = 0mm$

#### 3.2 `\abovetabulinesep` `\belowtabulinesep` and `\tabulinesep`

$$\begin{aligned}\backslash abovetabulinesep &= \langle \textit{dimen} \rangle \\ \backslash belowtabulinesep &= \langle \textit{dimen} \rangle\end{aligned}$$

`\abovetabulinesep` is the *minimal* vertical space allowed between the cell content and the cell top border. Setting this dimension to a positive value will insert – conditionally, depending on the cell content – a vertical space above the cell content.

`\belowtabulinesep` is conversely the minimal vertical space allowed to the bottom border of the cell. When `\belowtabulinesep > 0pt` a vertical space is inserted – conditionally, depending on the cell content – below the cell content.

3. `\arraystretch` is not a dimension but a macro that stores a scaling factor.

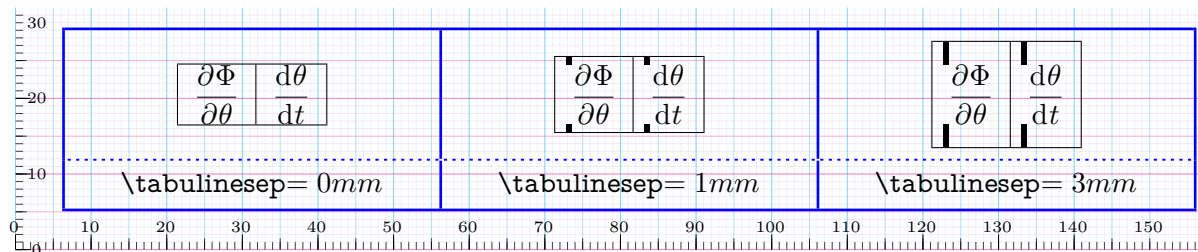
4. However `\extrarowsep` is not a dimension ! You can't test, for example, `\ifdim\extrarowsep > 0pt` ! Test `\extrarowheight` and `\extrarowdepth` instead, if needed.

`\tabulinesep=<dimen>`

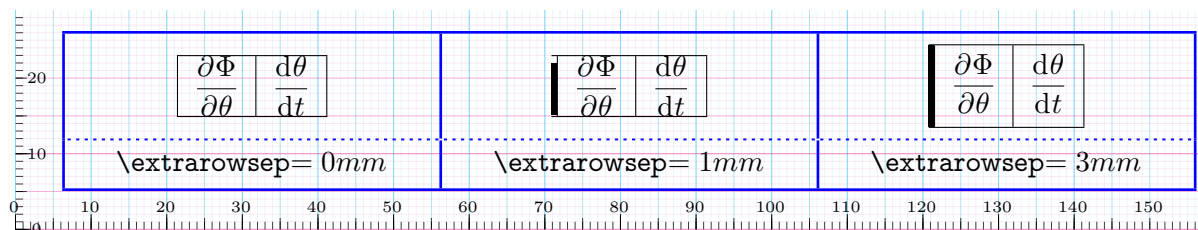
`\tabulinesep` makes the assignment for both `\abovetabulinesep` and `\belowtabulinesep`. The macro can be prefixed by `\global`<sup>5</sup>.

This parameter can be used in text and math modes to give more vertical space between lines, especially when using math formulae.

Examples (with `\tabustrutrule=2pt` to see the struts):



Now you can compare with `\extrarowsep`:



Presently (version 2.0), non zero value for `\tabulinesep` does not work properly with `\firstline` and `\lastline`, but you may use `\hline`, `\tabucline` and `booktabs` `\toprule`, `\midrule` and `\bottomrule`.

`\tabustrutrule=<dimen>`

For debugging purpose only (or to learn how this works): `\tabustrutrule` behaves like T<sub>E</sub>X `\overfullrule` and allows to see the struts introduced inside the cells to control the vertical spacing between lines. Struts are only inserted inside paragraph columns (p, m, b columns and also X columns which are paragraph columns as well).

Normally, those struts are rules of width 0pt.

## 4 Saving and restoring a tabu: `\savetabu`, `\usetabu` and `\preamble`

`\savetabu{<user-name>}`

The command `\savetabu` can be used at the end of any line of a `tabu` environment to save the target width (or the spread), the columns specifications (tabular preamble) and the widths of `tabu` X columns (if there is any). This possibility allows to easily make tabulars which share exactly the same shape throughout your document. This can also be used as a kind of `tabbing` environment which is able to remember the tabs positions.

If the `<user-name>` has been used before, an info is displayed in the `.log` file and the previous settings are overwritten.

With the `debugshow` package option (or `\tracingtabu`) the informations saved by `\savetabu` are reported in the `.log` file.

`\usetabu{<user-name>}`

`\usetabu` is the complement of `\savetabu`: it can be put alone in the `tabu` preamble instead of the usual columns specifications to restore any previous settings saved with `\savetabu`. The `<user-name>` must exist otherwise, you get an error. `\usetabu` is a help to **make several**

5. However `\tabulinesep` is not a dimension ! You can't test, for example, `\ifdim\tabulinesep>0pt` ! Test `\abovetabulinesep` and `\belowtabulinesep` instead, if needed.



**tabulars of exactly the same shape, same target, same preamble.** The only parameter that can be changed is the optional vertical position parameter for the whole tabular.

`\usetabu` does not work with `longtabu`.

`\usetabu` locally restores:

- the preamble<sup>6</sup>.
- the vertical position [c], [b] or [t], unless another position is specified.
- the target width of the `tabu` in points: the saved target width does not contain any control sequence: it is fixed and stored in points.
- the width of `tabu` X columns: those widths are not calculated any more, even in the case of negativ coefficients and X columns are directly transformed into p, m or b columns of the same widths as the ones that where calculated at the time of `\savetabu`
- `\tabcolsep` (or `\arraycolsep` in math mode) and `\arraystretch`
- `\arrayrulewidth`, `\doublerulesep`, `\extrarowheight`, `\extrarowdepth`, and `\extratabsurround`.
- `\minrowclearance`, `\arrayrulecolor` and `\doublerulesepcolor` (package `colortbl`)

`\abovetabulinesep` and `\belowtabulinesep` are not restored, because they are related to the content of the tabular rather than to its shape, and `\tabustrutrule` is not restored because of its “debugging” nature.

Example:

```
\extrarowheight=5pt\tabcolsep=12pt\arrayrulecolor{ForestGreen}
\begin{tabu}to .7\linewidth{XXX|X[c]|} \savetabu{mytabu} \tabucline*[on1pt]-
This & is & tabu & package \\ \tabucline*[on1pt]-
\end{tabu}
\arrayrulecolor{black}
```

This	is	tabu	package
------	----	------	---------

```
\begin{tabu}{\usetabu{mytabu}} \tabucline*[on1pt]-
\multicolumn{3}{c}{This is tabu} & package \\ \tabucline*[on1pt]-
\abovetabulinesep
\end{tabu}
```

This is tabu	package
--------------	---------

If one day you use `tabu`, you will have the idea to restore a `tabu` while modifying its target, or adding new columns... `\savetabu` and `\usetabu` have not been thought for this purpose, and you may have unexpected results.

`\preamble{⟨user-name⟩}`

`\preamble` can also be used after `\savetabu`. This is a variant of `\usetabu` that locally restores:

- the `tabu` (or `longtabu`) preamble.
- the vertical position [c], [b] or [t] (or [c], [l] or [r] for `longtabu`), unless another position is specified.
- the `tabu` / `longtabu` target width, unless another target is specified.

Any other tabular parameter is not restored.

Put `\preamble{⟨user-name⟩}` alone inside the `tabu` (or `longtabu`) preamble in place of the usual columns specifications.

`\preamble` works exactly as if you defined a `custom environment` for `tabu`.

6. The complete `\halign`-preamble is restored.

`\preamble` works with `longtabu`.

Example (continued...):

```
\begin{tabu} to\linewidth{\preamble{mytabu}} \tabucline[1pt off1pt]-
This      &is &tabu &package      \\\tabucline[1pt off1pt]-
\end{tabu}
```

This	is	tabu	package
------	----	------	---------

`\arrayrulecolor`, `\tabcolsep` *etc.* are not restored from `\savetabu`.

## 5 Differences between `tabu`, `tabular`, `tabularx` and `longtable`

### 5.1 Paragraph indentation

See [Paragraph indentation](#)

### 5.2 Custom environments

Unlike `tabularx`, it is possible to define your own environment using `tabu`:

```
\newenvironment{foo}
{\begin{tabu}{X[1.2]|[1pt gray]X}}
{\end{tabu}}
```

`tabu` environment, even when `X` columns are used, may appear in the definition of your custom `tabular` environment. You can also use the commands `\savetabu` and `\usetabu` for this purpose.

### 5.3 Inversion of tokens

When you typeset the following `tabular`:

```
\begin{tabular}{|>{\bfseries}>{ before }l<{ one }<{ two }|}
cell content
\end{tabular}
```

You get the following result: 

before	<b>cell content</b>	two	<b>one</b>
--------	---------------------	-----	------------

→ The word *before* is not bold, and **two** comes before **one**.

The reason is explained in the documentation of `array.sty`, and is related to the `array` environment in math mode when using `\newcolumntype`.

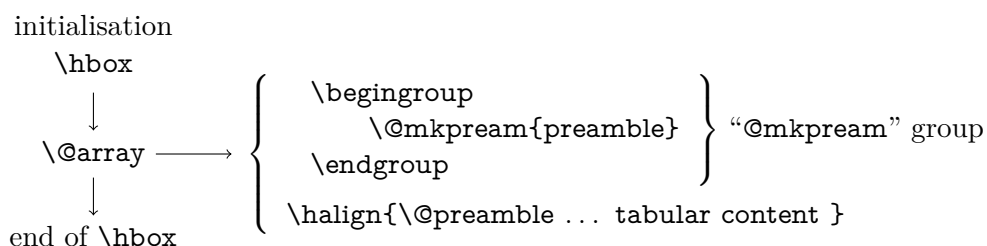
This rather strange inversion of tokens may be justified in math mode (otherwise, errors may occur) but not in text mode in our opinion. Inside a `tabu` environment, when not in math mode, the tokens are not reversed and you get the intuitively expected result:

before	cell content	one	two
--------	--------------	-----	-----

In math mode however, tokens are in the reverse order in the `tabu` environment like they are in the `array` environment.

### 5.4 Improved process for rewriting columns (*for keen readers*)

Any `tabular` that does not split accross pages is made with the following process:



For more details, see the [Flow chart of expansion](#).

`\@mkpream` works in two times inside a (semi-simple) group:

*First the rewriting process:*

Each special column in the tabular preamble is transformed into one the columns defined by `array.sty`.

*Second the building of the `\halign` preamble:*

The “rewritten preamble” is parsed and transformed in a preamble for the T<sub>E</sub>X primitive `\halign`. The result is stored into the `\@preamble` macro.

Any special columns of `tabu` are defined only inside the “`@mkpream`” group.

In the following example, you get an error with `tabular` and no error with `tabu`. With `tabular`, and `siunitx` `S` column, the *rewritting process* is as follow:

	Inside <code>tabular</code> :
	1) Rewrite <code>S</code> : not found because inside <code>{...}</code>
<code>\documentclass{minimal}</code>	2) Rewrite <code>*</code>
<code>\usepackage{numprint,siunitx,xcolor}</code>	3) Rewrite <code>n</code> column defined by package <code>numprint</code>
<code>\usepackage{tabu}</code>	Then the ‘n’ in <code>green</code> is rewritten
<code>\begin{document}</code>	→ problem
<code>\begin{tabular}{*2{S[color=green]}}</code>	
<code>123,45</code>	Inside <code>tabu</code> :
<code>\end{tabular}</code>	1) Rewrite <code>*</code>
	2) Rewrite <code> </code> (there is none here)
<code>\begin{tabu}{*2{S[color=green]}}</code>	go back
<code>123,45</code>	3) Rewrite <code>*</code>
<code>\end{tabu}</code>	4) Rewrite <code> </code>
	5) Rewrite <code>S</code>
<code>\end{document}</code>	6) Rewrite <code>n</code> → not found because <code>S</code> was rewritten before, according to <code>siunitx</code> definition.

The process of rewriting columns is usually longer inside `tabu` than inside `tabular`, but conversely `tabu` with `X` columns is optimised compared to `tabularx`, because the preamble is built only once, and not rebuilt before each trial as `tabularx` does. Thus `tabu` is much quicker than `tabularx`.

The process of rewriting is very sensitiv to the order in which columns are actually rewritten. This becomes critical when columns are defined with an optional argument like `tabu X` and `|` columns or `siunitx S` column.

If it possible to define a new column types using the `X` token for use with `tabu`:

```
\newcolumntype{C}{X[c]}
```

it is not recommanded no nest such constructions like:

```
\newcolumntype{Q}{>{\color{green}}C}
```

In fact, a problem may arise in nested `tabus` if such a `Q` column is defined before the `C` column...

Well, just avoid to nest new column types definitions !

## 6 The package options

### 6.1 The debugshow package option

```
\tracingtabu
\tracingtabu=2
```

The control sequence `\tracingtabu` has the same effect as the `debugshow` option:

- `tabu` will report the widths it computes at each attempt to read the target, when `X` columns are used.
- Saved informations on the `tabu` are reported in the `.log` file when `\savetabu` is used.

Typical information in the `.log` file:

```
(tabu) Try      tabu X      tabu Width      Target      Coefs      Update
(tabu)  1)  386.67296pt    797.34592pt    386.67296pt    2.0pt    -205.33649pt
(tabu)  2)  181.33647pt    386.67294pt    386.67296pt    2.0pt     0.00002pt
(tabu)  2)  Target reached (hfuzz=0.1pt) *****
```

What does it mean?

- 1) The first attempt was performed with `X=386.67296pt`  
The `tabu` width (797.34592pt) exceeded the target by 410.67296pt.  
Thus `X` has been updated:  $410.67296\text{pt} / 2 = 205.33649\text{pt}$  and then:  
 $X = 386.67296\text{pt} - 205.33649\text{pt} = 181.33647\text{pt}$
- 2) The second attempt lead to a `tabu` width of 386.67294pt: the target is reached.  
The final width of each `X` column is the product of `tabu X` by its width coefficient.

`\tracingtabu=2` gives even more information about the measuring of natural widths of columns.

### 6.2 The linegoal package option

With the `linegoal` option, the default target for `tabu` with `X` columns is `\linegoal` instead of `\linewidth`. The `linegoal` package must be loaded and compilation must be done with pdf<sub>T</sub>E<sub>X</sub>, otherwise, a warning is displayed and the `linegoal` option has no effect: the default target remains `\linewidth`. `\linegoal` works with pdf<sub>T</sub>E<sub>X</sub> in pdf mode **and in dvi** mode.

If for some reason, you wish to turn down the `linegoal` option in your document, you can say (in a group for example): `\let\tabudefaulttarget=\linewidth`

In any case, specifying the target overwrites the default: `\begin{tabu} to\linewidth`

### 6.3 The delarray package option

`delarray` option has the single effect to load `delarray.sty` for delimiters shortcuts around `tabu`. Delimiters shortcuts work both in math and text mode.

### 6.4 The light package option

When you enter a `tabu` environment, two macros amongst the smallest possible of `array.sty` are modified: these are `\prepnext@tok`, which is expanded while `\@mkpream` builds the `\halign` preamble and `\save@decl` to avoid inversion of tokens in text mode (see [Inversion of tokens](#)).

Modification of `\prepnext@tok` is loaded only inside the `tabu` environment, while the modification of `\save@decl` is loaded only inside the group in which `\@mkpream` works: this is very very local to `tabu` and cannot interfere with any other tabular !

`\prepnext@tok` could have been loaded inside the `\@mkpream` group as well, but then `\rowfont` would not have worked inside `array` or `tabular` nested into a `tabu`. Thus the choice.

As you see, these modifications do no modify `tabular`, `tabular*`, `tabularx`, `longtable` *etc.etc.etc.*

If for some reason you prefer the original macros of `array.sty`, then you can load `tabu` with the “light” option. As a consequence: `\rowfont` and `\tabucline` will not be available, and vertical

lines | will not have an optional argument for their widths and color. However, `tabu X` columns will work normally.

## 7 Corrections of some bugs (*available only inside tabu*)

### 7.1 Correction for `colortbl` and `arydshln`: compatibility with `delarray`

Both `colortbl` and `arydshln` forget the control sequence `\@arrayright` in their implementation, quite strangely because both of them take care of `\@arrayleft`. As a result, `delarray` shortcuts for delimiters around a tabular does not work if `colortbl` and/or `arydshln` are loaded.

Those control sequences are used by the `delarray` package to put variable size delimiters around the array:

<code>\begin {tabu}\{{X}.</code>		<code>\left \{ \begin {tabu}{X}</code>
<code>...</code>	is like:	<code>...</code>
<code>\end {tabu}</code>		<code>\end {tabu} \right .</code>

### 7.2 Correction for `arydshln`: `@` columns

A bug in `\adl@xarraydashrule: !-arg` columns (class 1) and `@-arg` columns (class 5) should be treated the same as far as rules are concerned.

With this correction, the “known problem number 1” in `arydshln` documentation is solved.

## 8 To do for even better `tabus`

In decreasing order of priority:

- ➡ Implement `\tabulineskiplimit` in a manner close to `tbls \tabulinesep` or `cellspace S` columns-modifier.
- ➡ Make double `\tabucline` compatible with `colortbl \doublerulesepcolor`
- ➡ Multiple `\tabucline` between different columns: extended specs:  
`\tabucline[(line spec)]{start-stop, start-stop}[(line spec)]{start-stop} ...`
- ➡ Presently, `longtabu` with `X` columns works only if `\LTchunksize` is greater than the number of rows. I compiled a `longtabu` of 56 pages on my PC with `\LTchunksize = 2 000` without problem. Presently `\LTchunksize` is set to 10 000 during trials when `longtabu` contains `X` columns.
- ➡ Make `\tabucline` work with page breaks (one line on the top of the page, one line on the bottom of the previous).
- ➡ An environment “`calstabu`” based on the `cals` package but with the syntax of `tabular` if possible...

## 9 Implementation

### 9.1 Identification, requirements and options

The package namespace is `tabu@`.

```

1 \*package
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{tabu}[2011/01/25 v2.0 - flexible LaTeX tabulars (FC)]
4 \RequirePackage{array}[2008/09/09]
5 \RequirePackage{varwidth}[2009/03/30]

```

Minimal catcode acertaining for loading `tabu` in good conditions:

```

6 \AtEndOfPackage{\tabu@AtEnd\let\tabu@AtEnd\undefined}
7 \let\tabu@AtEnd\empty
8 \def\TMP@EnsureCode#1#2{%
9   \edef\tabu@AtEnd{%
10     \tabu@AtEnd
11     \catcode#1 \the\catcode#1\relax
12   }%
13   \catcode#1 #2\relax
14 }% \TMP@EnsureCode
15 \TMP@EnsureCode{33}{12} % !
16 \TMP@EnsureCode{124}{12}% |
17 \TMP@EnsureCode{0}{12}% ^~@
18 \TMP@EnsureCode{36}{3}% $ = math shift

```

`\tracingtabu` (debugshow option) `\tracingtabu` is the same as option `debugshow`.

```

19 \def\tracingtabu{\begingroup\@ifnextchar=%
20   {\afterassignment\tabu@tracing\count@}
21   {\afterassignment\tabu@tracing\count@1\relax}}
22 \def\tabu@tracing{\expandafter\endgroup
23   \expandafter\tabu@tr@cing\number\count@\@nil
24 }% \tabu@tracing
25 \def\tabu@tr@cing #1\@nil{%
26   \ifnum#1>\z@
27     \let\tabu@message \message \let\tabu@message@save \tabu@tracing@save
28     \ifdefined \pdfelapsedtime
29       \def\tabu@timerstart {\edef\tabu@starttime{\the\pdfelapsedtime}}%
30       \let\tabu@elapsedtime \tabu@message@etime \fi
31     \ifnum#1>\@ne \let\tabu@debug \message
32     \else \let\tabu@debug \@gobble \fi
33   \else
34     \let\tabu@message \@gobble
35     \let\tabu@message@save \@gobble
36     \let\tabu@debug \@gobble
37     \let\tabu@timerstart \relax
38   \fi
39 }% \tabu@tr@cing
40 \tracingtabu=\z@
41 \DeclareOption{debugshow}{\AtEndOfPackage{\tracingtabu=2\relax}}

```

 `linegoal` (package option)

```

42 \let\tabubefaulttarget \linewidth
43 \DeclareOption{linegoal}{%
44   \AtEndOfPackage{%
45     \RequirePackage{linegoal}[2010/12/07]%
46     \let\tabubefaulttarget \linegoal% \linegoal is \linewidth if not pdfTeX
47   }}

```



`delarray` (package option)

```
48 \DeclareOption{delarray}{%
49   \AtEndOfPackage{\RequirePackage{delarray}}%
50 }
```

`light` (package option)

```
51 \DeclareOption{light}{%
52   \AtEndOfPackage{%
53     \let\tabu@prepnext@tok \prepnext@tok   \let\tabu@save@decl \save@decl
54     \let\tabu@rowfont      \tabu@norowfont \let\tabu@cline      \tabu@nocline
55     \let\tabu@firstcline   \relax          \let\tabu@lines      \relax
56     \let\tabu@setup@everycr\relax          \let\tabu@multicolumn \multicolumn
57   }
58 }
59 \def\tabu@norowfont{\PackageError{tabu}
60   {\string\rowfont\space is not available with option ‘light’}\@ehd}
61 \def\tabu@nocline{\PackageError{tabu}
62   {\string\tabu@cline\space is not available with option ‘light’}\@ehd}

63 \ProcessOptions
```

At Begin Document, a fix for `arydshln` and `colortbl` compatibility with `delarray` shortcuts available inside `tabu`: requirement for this fix is checked by `\tabu@fix@arrayright`.

Then the switch `\iftabu@colortbl` is set.

Finally, `\longtabu` is defined only if the `longtable` package is detected.

```
64 \AtBeginDocument{%
65   \expandafter\in@ \expandafter\@arrayright\expandafter{\endarray}%
66   \ifin@ \let\tabu@endarray \endarray
67   \else \tabu@fix@arrayright \fi % <fix for colortbl & arydshln (delarray)>
68   \@ifpackageloaded{colortbl} \tabu@colortbltrue \tabu@colortblfalse
69   \@ifpackageloaded{arydshln}{%
70     {\let\tabu@adl@fix \relax \let\tabu@adl@endTRIAL \empty}%
71     \@ifpackageloaded{longtable}{\let\longtabu \tabu@nolongtabu}%
72     \@ifpackageloaded{cellspace} \tabu@warn@cellspace \relax
73 }
74 \def\tabu@warn@cellspace{%
75   \PackageWarning{tabu}{%
76     Package cellspace has some limitations \MessageBreak
77     And redefines some macros of array.sty.\MessageBreak
78     Please use \string\tabulinesep\space to control\MessageBreak
79     vertical spacing of lines inside tabu environnement}%
80 }% \tabu@warn@cellspace
```

## 9.2 Some constants

<code>\tabu@cnt</code>	Used in <code>\tabu@arith</code> (the number of trials) and <code>\tabu@prepnext@tok</code> (for <code>\rowfont</code> ).
<code>\tabu@nbc</code>	A counter that save the number of columns of the <code>tabu</code> .
<code>\tabu@X@cols</code>	Used only when <code>tabu X</code> columns are used with “ <code>tabu spread</code> ”.
<code>\tabu@nested</code>	A counter that stores the deepness of nested <code>tabu</code> ’s.
<code>\tabu@start</code> <code>\tabu@stop</code>	Counters used for <code>\tabu@cline</code> to get the starting and ending columns.
	<pre>81 \newcount \tabu@cnt 82 \newcount \tabu@nbc 83 \newcount \tabu@X@cols 84 \newcount \tabu@nested 85 \newcount \tabu@start</pre>

	86 \newcount \tabu@stop	
\tabu@target	Stores the tabu target (either “to” or “spread”).	
\tabu@spreadtarget	Used only when tabu X columns are used with “tabu spread”, or with negativ coefficients.	
\tabu@naturalX	87 \newdimen \tabu@target 88 \newdimen \tabu@spreadtarget 89 \newdimen \tabu@naturalX	
\tabucolX	The dimen corresponding to the preamble token X[1]: the standard width of X columns.	
\tabu@X@sum	Stores the sum of all width coefficients for X columns.	
\tabucolX@error	In case of X columns too narrow. 90 \newdimen \tabucolX 91 \newdimen \tabu@X@sum 92 \newdimen \tabucolX@error \tabucolX@error=1em	
\abovetabulinesep	The two dimensions \abovetabulinesep and \belowtabulinesep allow to control vertical spacing	
\belowtabulinesep	of lines inside a tabu in a way similar to the philosophy of <a href="#">cellspace</a> . However, limitations of	
\extrarowdepth	cellspace are lifted (nested tabu environments, use of colors... see the <a href="#">cellspace limitations</a> in the	
\tabustrutrule	revision history).  The dimen \tabustrutrule is here only for debugging purpose: its value must be 0pt. It behaves like T <sub>E</sub> X \overfullrule to see the struts introduced to control the vertical spacing of lines. 93 \newdimen \abovetabulinesep 94 \newdimen \belowtabulinesep 95 \newdimen \extrarowdepth 96 \newdimen \tabustrutrule \tabustrutrule \z@	
\iftabu@measuring	This switch is set to true by \tabu@arith if the trial did not reached the target.  It is also temporarily set to true when the first X column is encountered in the tabu preamble, at the time \@mkpream scans it to built the \halign preamble. The first X column found actually triggers some special setup to be expanded before \halign (see the <a href="#">flow chart...</a> )	
\iftabu@spread	A switch whether “tabu spread” is used or not.	
\iftabu@firstcline	A switch to adapt \tabucline automatically if it comes first in the tabu (similar to \firsthline).	
\iftabu@negcoef	A switch set to true in case of negativ coef (natural width if less than X[coef]). 97 \newif \iftabu@measuring 98 \newif \iftabu@spread 99 \newif \iftabu@firstcline 100 \newif \iftabu@negcoef	
\tabu@box	Stores the whole tabu when an attempt to adjust X columns is performed.  It is also used by \tabucline to save the \@arstrutbox when inserting a horizontal line. 101 \newsavebox \tabu@box 102 \chardef \tabu@arstrutbox=\@arstrutbox 103 \let\tabu@vcenter \vcenter	
\tabu@gobblespace	A macro which is needed when scanning tokens with \futurelet.	
\tabu@gobbleopt	A macro that gobbles an optional argument. 104 \def\tabu@gobblespace#1 {#1} 105 \def\tabu@gobbleopt#1{\@ifnextchar [{\tabu@gobble@pt{#1}}{#1}}% 106 \def\tabu@gobble@pt#1[#2]{#1}	

`\tabu@save@decl` No inversion on tokens in the tabu preamble, when not in math mode.

```
107 \def\tabu@save@decl{%
108   \toks\count@ =\expandafter{\the\toks\expandafter\count@ \@nextchar}%
109 }% \tabu@save@decl
```

`\tabu@newcolumnntype` A helper macro to create new column types for tabu.

The column types are not appended to `\NC@list` in order to keep them local to tabu.

`\tabu@privatecolumnntype` Columns types defined with `\tabu@privatecolumnntype` are "mounted" only inside the `\@mkpream` group of tabu.

```
110 \def\tabu@newcolumnntype #1{%
111   \expandafter\tabu@new@columnntype
112     \csname NC@find@\string#1\expandafter\endcsname
113     \csname NC@rewrite@\string#1\endcsname
114     {#1}%
115 }% \tabu@newcolumnntype
116 \def\tabu@new@columnntype #1#2#3{%
117   \def##1##1#3{\NC@{##1}}%
118   \let#2\relax \newcommand*#2%
119 }% \tabu@new@columnntype
120 \def\tabu@privatecolumnntype #1{%
121   \expandafter\tabu@private@columnntype
122     \csname NC@find@\string#1\expandafter\endcsname
123     \csname NC@rewrite@\string#1\expandafter\endcsname
124     \csname tabu@NC@find@\string#1\expandafter\endcsname
125     \csname tabu@NC@rewrite@\string#1\endcsname
126     {#1}%
127 }% \tabu@privatecolumnntype
128 \def\tabu@private@columnntype#1#2#3#4{%
129   \g@addto@macro\tabu@privatecolumns{\let#1#3\let#2#4}%
130   \tabu@new@columnntype#3#4%
131 }% \tabu@private@columnntype
132 \let\tabu@privatecolumns \@empty
```

**X (private column type)** This is the rewrite macro for tabu X columns. Such a column has an optional argument: the width coefficient for the tabu X column whose default value is 1, and may be some alignments parameters. The coefficient is used in the expression: `p{\dimexpr<coef>\tabucolX}`

```
133 \tabu@privatecolumnntype X[1] []{\tabu@rewrite@X{#1}}%
134   \let\@halignto \relax
135   \expandafter \NC@find \tabucolX@rw}
```

The next part of the definition (`\tabu@rewrite@X`) can be found page 33.

**\usetabu (private column type)** `\usetabu` is defined as a tabu new column type: loaded only inside the `\@mkpream` group inside the tabu environment.

```
136 \tabu@privatecolumnntype \usetabu [1]{%
137   \ifx\\#1\\ \tabu@saveerr{}\else
138     \@ifundefined{tabu@saved@\string#1}
139       {\tabu@saveerr{#1}}
140       {\let\tabu@rewrite@X \tabu@rewrite@Xrestore
141         \csname tabu@saved@\string#1\expandafter\endcsname\expandafter\z}%
142   \fi
143 }% \NC@rewrite@\usetabu
```

`\preamble` (private column type) `\preamble` is defined as a `tabu` new column type: loaded only inside the `\@mkpream` group inside the `tabu` environment.

```

144 \tabu@privatecolumnntype \preamble [1]{%
145   \ifx\#1\\\tabu@saveerr{}\else
146     \ifundefined{tabu@saved@\string#1}
147       {\tabu@saveerr{#1}}
148       {\csname tabu@saved@\string#1\expandafter\endcsname\expandafter\@ne}%
149   \fi
150 }% \NC@rewrite@\preamble

```

`\tabu@rewritefirst` (new column type) This new column type is not really a column type! It is always added to a `tabu` preamble in order to do some setup before any other column is rewritten by `\@mkpream`.

Thus, `\NC@do\tabu@rewritefirst` is added **at the beginning of** `\NC@list` at the entry of a (not nested) `tabu` environment.

This “column type” sets up the new column type `\tabu@rewritelast` which is added to **at the end of** `\NC@list`, and defines the token `X` to be rewritten by `\tabu@NC@rewrite@X` (in case `tabularx` is used with `tabu`, this modification of the `X` column occurs only inside the group where `\@mkpream` does its job).

```

151 \tabu@newcolumnntype \tabu@rewritefirst{%
152   \tabu@X@cols \z@
153   \let\tabu@dimexpr \relax
154   \let\tabu@savepreamble \relax
155   \let\NC@rewrite@X \tabu@NC@rewrite@X
156   \gdef\tabu@global@temp {\tabu@savepreamble}%
157   \aftergroup \tabu@global@temp
158   \let\tabu@islast \relax
159   \tabu@privatecolumns
160   \NC@list{\NC@do\usetabu \NC@do\preamble \NC@do *}%
161   \tabu@lines % defines NC@rewrite@ for tabu only (inside @mkpream group)
162   \NC@list\expandafter{\the\expandafter\NC@list \expandafter\NC@do \expandafter X%
163     \tabu@NC@list \NC@do X\NC@do\tabu@rewritemiddle}%
164   \ifcat$\dollar\end\else
165     \let\save@decl \tabu@save@decl \fi % no inversion of tokens in text mode
166   \NC@find \tabu@rewritemiddle
167 }% \NC@rewrite@\tabu@rewritefirst
168 \def\tabu@savepreamble{%
169   \NC@list\expandafter{\tabu@NC@list}%
170   \let\tabu@savedpreamble \@preamble
171   \ifdim \tabustrutrule>\z@
172     \setbox\@arstrutbox\hbox{\vrule
173       \@height \ht\@arstrutbox
174       \@width \tabustrutrule
175       \@depth \dp\@arstrutbox}%
176   \fi
177   \ifdim \extrarowdepth>\z@
178     \setbox\@arstrutbox\hbox{\vrule
179       \@height \ht\@arstrutbox
180       \@width \tabustrutrule
181       \@depth \dimexpr\dp\@arstrutbox+\arraystretch\extrarowdepth}%
182   \fi
183 }% \tabu@savepreamble

```

`\tabu@rewritelast` (new column type) This new column type is rewritten after  $X$  columns, because it is declared by when the column `\tabu@rewritefirst` is actually rewritten. In the case where `\tabu@target` is  $> 0$  (either because of “`tabu to`” or “`tabu spread`” has been called) and if there is no  $X$  column, then `@{\extracolsep\@flushglue}` is added at the beginning of the preamble.

To avoid duplicate margin in the `tabu` we have to test the next token in the preamble. If the next token is `|` or `!` then no margin must be added and `@{\extracolsep\@flushglue}` can be inserted at the beginning of the preamble.

Otherwise, we must insert `!{\extracolsep\@flushglue}` in order to keep the margin.

`\tabu@rewritelast` column type is loaded by `\tabu@rewritefirst` column type, only inside the `\@mkpream` group inside the `tabu` environment.

```

184 \tabu@newcolumnntype \tabu@rewritemiddle{%
185   \NC@list\expandafter{\tabu@NC@list \NC@do\tabu@endrewrite \NC@do\tabu@rewritelast}%
186   \tabu@checklast
187 }% \NC@rewrite@\tabu@rewritemiddle
188 \def\tabu@checklast#1\relax{%
189   \def\tabu@temp{#1}%
190   \ifx \tabu@temp\tabu@islast
191     \expandafter\tabu@lastrewrite
192   \else \let\tabu@islast \tabu@temp
193     \expandafter\NC@find \expandafter\tabu@rewritelast
194   \fi
195   #1\relax
196 }% \tabu@checklast
197 \tabu@newcolumnntype \tabu@rewritelast{\NC@find \tabu@endrewrite}
198 \tabu@newcolumnntype \tabu@endrewrite{\tabu@checklast}
199 \def\tabu@lastrewrite{%
200   \let\tabu@next \NC@find
201   \ifx \@halignto\relax % found a X column
202     \edef\tabu@global@X{\tabu@global@X \tabu@X@cols\the\tabu@X@cols\relax}%
203     \ifx \tabu@halignto\relax % X column measure
204       \tabu@global@Xnested
205       \iftabu@negcoef \aftergroup\tabu@negcoeftrue \fi
206       \aftergroup \tabu@prep@TRIAL
207     \else \aftergroup \tabu@setup@everycr
208     \fi
209   \else
210     \aftergroup \tabu@setup@everycr
211     \ifdim \tabu@target=\z@ \else
212       \let\tabu@next \tabu@extracolsep \fi % <need \extracolsep ?>
213   \fi
214   \let\@halignto \tabu@halignto
215   \tabu@next
216 }% \tabu@lastrewrite
217 \def\tabu@extracolsep{\futurelet\tabu@temp \tabu@extracol@sep}
218 \def\tabu@extracol@sep{%
219   \ifx \tabu@temp\@sptoken
220     \expandafter\tabu@gobblespace \expandafter\tabu@extracol@sep
221   \else
222     {\endlinechar\m@ne\scantokens{\let\x |}}%
223     \expandafter\ifx \x\tabu@temp
224       \def\tabu@next {\NC@find @{\extracolsep\@flushglue}}\else
225     {\endlinechar\m@ne\scantokens{\let\x !}}%
226     \expandafter\ifx \x\tabu@temp
227       \def\tabu@next {\NC@find @{\extracolsep\@flushglue}}\else
228       \def\tabu@next {\NC@find !{\extracolsep\@flushglue}}\fi\fi
229     \expandafter\tabu@next

```

```

230 \fi
231 }% \tabu@extracol@sep
232 \def\tabu@global@Xnested{\expandafter\tabu@global@X@nested
233 \csname tabu@global@X@the\tabu@nested\endcsname}
234 \def\tabu@global@X@nested#1{%
235 \global\let#1\tabu@global@X
236 \aftergroup\let \aftergroup\tabu@global@X \aftergroup#1%
237 }% \tabu@global@Xnested

```

`\tabu@rewrite@multicolumn` A special rewrite to allow `[[...]]` in `\multicolumn` preamble inside `tabu` environment.

```

238 \tabu@newcolumnntype \tabu@rewrite@multicolumn{%
239 \aftergroup\endgroup % after \mkpream group (restores \multispan)
240 \tabu@lines
241 \ifcat$\dollar\end\else
242 \let\save@decl \tabu@save@decl \fi % no inversion of tokens in text mode
243 \NC@find
244 }% \tabu@rewrite@multicolumn
245 \def\tabu@multicolumn{%
246 \expandafter\ifx\csname\@currentenv\endcsname\tabu
247 \expandafter\tabu@multicolumn
248 \else \expandafter\tabu@multicolumnORI
249 \fi
250 }% \tabu@multicolumn
251 \def\tabu@multicolumn#1#2{\multispan{#1}\begingroup
252 \NC@list{\NC@do \tabu@rewrite@multicolumn}\let\multispan \@gobble
253 \tabu@multicolumnORI{#1}{\tabu@rewrite@multicolumn #2}%
254 }% \tabu@multicolumn

```

`\iftabu@colortbl` The switch `\iftabu@colortbl` is used by `\rowfont` when modifying the alignment, because `colortbl` changes the glue put inside the `\halign` preamble to make standard alignments. This switch is set At Begin Document.

```
255 \newif\iftabu@colortbl
```

`\tabu@nowrite` A trick (from the T<sub>E</sub>X-book) to forbid `\write` when a trial is done on the `\halign`.

```

256 \def\tabu@nowrite#1#2{\afterassignment}\toks@}
257 \let\tabu@write\write
258 \let\tabu@immediate\immediate
259 \protected\def\tabu@GenericError{\begingroup
260 \def\immediate\write{\aftergroup\endgroup
261 \tabu@immediate\tabu@write}\tabu@GenericErrorORI}%
262 \let\tabu@GenericErrorORI \GenericError
263 \protected\def\tabu@warn{\begingroup
264 \def\immediate\write{\aftergroup\endgroup
265 \tabu@immediate\tabu@write}\PackageWarning{tabu}}

```

### 9.3 `\tabu`, `\endtabu`, `\longtabu` and `\endlontabu`

`\tabu` `\tabu` is the command of the environment.

`\endtabu` `\endtabu` is `\endtabular` or `\endarray` in math mode.

```

266 \def\tabu{%
267 \ifmmode \def\tabu@{\array}\def\endtabu{\endarray}%
268 \else \def\tabu@{\tabular}\def\endtabu{\endtabular}\fi
269 \let\tabu@arrayleft@measure \tabu@arrayleftmeasure
270 \tabu@setup \tabu@settarget
271 }% \tabu
272 \def\longtabu{%
273 \ifmmode\PackageError{tabu}{\longtabu not allowed in math mode}\fi

```



```

274 \def\tabu@{\longtable}\def\endlongtabu{\endlongtable}%
275 \let\tabu@arrayleft@measure \tabu@long@arrayleft@measure
276 \tabu@setup \tabu@settarget
277 }% \longtabu
278 \def\tabu@nolongtabu{\PackageError{tabu}
279 {longtabu requires the longtable package}\@ehd}

```

`\tabu@setup` This macro sets the `tabu` X column definition at the beginning of the `tabu` environment.

```

280 \def\tabu@setup{%
281 \tabu@adl@fix \let\endarray \tabu@endarray % <fix> colortbl & arydshln (delarray)
282 \let\tabu@global@X \@empty
283 \ifx \verbatim\tabu@sanitizetext % <nested tabu>
284 \tabu@setup@saveglobal
285 \advance\tabu@nested \@ne
286 \def\tabu@aligndefault{t}\let\tabudefaulttarget \linewidth
287 \else \tabu@timerstart
288 \let\tabu@multicolumnORI \multicolumn
289 \let\multicolumn \tabu@multicolumn
290 \def\tabu@aligndefault{c}\edef\tabu@hfuzz {\the\hfuzz}%
291 \global\tabu@footnotes{}\aftergroup\the\aftergroup\tabu@footnotes
292 \ifdim\parindent>\z@ \ifx\linewidth\tabudefaulttarget
293 \everypar\expandafter{% % correction for indentation
294 \the\everypar\everypar\expandafter{\the\everypar}%
295 \setbox\z@=\lastbox
296 \ifdim\wd\z@>\z@ \advance\linewidth -\wd\z@\fi
297 \box\z@
298 }%
299 \fi\fi
300 \fi
301 \ifcase 0\ifdim\abovetabulinesep>\z@ 1\else\ifdim\belowtabulinesep>\z@ 1\fi\fi
302 \let\tabu@celllalign \@empty \let\tabu@cellralign \@empty
303 \else \def\tabu@celllalign{\tabu@verticalmeasure}%
304 \def\tabu@cellralign{\tabu@verticalspacing}%
305 \fi
306 \let\@footnotetext \tabu@footnotetext \let\@xfootnotetext \tabu@xfootnotetext
307 \let\@xfootnote \tabu@xfootnote \let\centering \tabu@centering
308 \let\raggedright \tabu@raggedright \let\raggedleft \tabu@raggedleft
309 \tabu@trivlist %<restore \=\@normalcr inside lists>
310 \def\tabudecimal {\tabu@tabudecimal}\let\verbatim \tabu@sanitizetext
311 \let\fbbox \tabu@fbbox \let\color@b@x \tabu@color@b@x
312 \let\rowfont \tabu@rowfont
313 \let\prepnext@tok \tabu@prepnext@tok %<for \rowfont and \tabucline>
314 \everycr{}\let\tabu@everycr \everycr
315 \tabu@spreadfalse \tabu@measuringfalse \tabu@negcoeffalse
316 \edef\tabu@NC@list{\the\NC@list}\NC@list{\NC@do \tabu@rewritefirst}%
317 }% \tabu@setup
318 \def\tabu@setup@saveglobal{%
319 \expandafter\xdef\csname tabu@global@\the\tabu@nested\endcsname{%
320 \global\tabu@X@cols \the\tabu@X@cols\relax
321 \global\tabu@naturalX \the\tabu@naturalX\relax
322 \gdef\noexpand\tabu@naturalX@min {\tabu@naturalX@min}%
323 \gdef\noexpand\tabu@naturalX@max {\tabu@naturalX@max}}%
324 \expandafter\aftergroup\csname tabu@global@\the\tabu@nested\endcsname
325 }% \tabu@setup@saveglobal

```

`\tabu@settarget` The macro sets `\tabu@target` (a dimen) to the value specified for “`tabu to`” or “`tabu spread`”.

```

\tabu@begin
326 \def\tabu@settarget{\futurelet\@let@token \tabu@sett@rget}
327 \def\tabu@sett@rget{\tabu@target\z@
328   \ifcase \ifx \bgroup\@let@token 0\else
329     \ifx [\@let@token 0\else
330     \ifx \@sptoken\@let@token 1\else
331     \if t\@let@token 2\else
332     \if s\@let@token 3\else
333     \m@ne\fi\fi\fi\fi\fi\relax
334   \expandafter\tabu@begin
335   \or \expandafter\tabu@gobblespace\expandafter\tabu@settarget
336   \or \expandafter\tabu@to
337   \or \expandafter\tabu@spread
338   \else \expandafter\tabu@begin
339   \fi
340 }% \tabu@sett@rget
341 \def\tabu@to to{\def\tabu@halignto{to}\tabu@gettarget}
342 \def\tabu@spread spread{\tabu@spreadtrue\def\tabu@halignto{spread}\tabu@gettarget}
343 \def\tabu@gettarget {\afterassignment\tabu@linegoaltarget \tabu@target}
344 \def\tabu@linegoaltarget {\futurelet\tabu@temp \tabu@linegoal@target}
345 \def\tabu@linegoal@target {%
346   \ifx \tabu@temp\LNGL@setlinegoal
347     \LNGL@setlinegoal \expandafter \@firstoftwo \fi % @gobbles \LNGL@setlinegoal
348   \tabu@begin
349 }% \tabu@linegoal@target
350 \def\tabu@begin #1#{%
351   \edef\tabu@halignto{\ifdim\tabu@target>\z@ \tabu@halignto\the\tabu@target\fi}%
352   \@testopt \tabu@@begin \tabu@aligndefault #1\@nil
353 }% \tabu@begin
354 \def\tabu@@begin [#1]#2\@nil#3{%
355   \tabu@preparesave{#3}% <for \savetabu>
356   \def\tabu@align {#1}%
357   \tabu@ [\tabu@align]#2{\tabu@rewritefirst #3}%
358 }% \tabu@@begin

```

`\tabu@preparesave` This macro temporarily saves the `tabu` preamble and the settings for `\arrayrulewidth`, `\extrarowheight` *etc.*.

Those values are retrieved by `\savetabu`, if invoked inside the `tabu`.

```

359 \def\tabu@preparesave #1{%
360   \iftabu@colortbl \temptokena{}%
361   \tabu@prepare@save \CT@arc@
362   \tabu@prepare@save \CT@drsc@
363   \fi
364   \edef\tabu@sav@param{%
365     \ifmmode \col@sep \the\arraycolsep
366     \else \col@sep \the\tabcolsep \fi\relax
367     \arrayrulewidth \the\arrayrulewidth\relax
368     \doublerulesep \the\doublerulesep\relax
369     \extrarowheight \the\extrarowheight\relax
370     \extrarowdepth \the\extrarowdepth\relax
371     \extratabsurround \the\extratabsurround\relax
372     \def\noexpand\arraystretch {\arraystretch}%
373     \iftabu@colortbl \minrowclearance\the\minrowclearance\relax
374     \the\temptokena
375   \fi}%
376   \def\tabu@sav@pream {\edef\tabu@halignto{to\the\tabu@target}\NC@find #1}%
377 }% \tabu@preparesave

```

```

378 \def\tabu@prepare@save #1{%
379   \ifx\relax#1\else
380     \@temptokena\expandafter{\the\expandafter\@temptokena
381       \expandafter\def\expandafter#1\expandafter{#1}}\fi
382 }% \tabu@prepare@save

```

## 9.4 Footnotes inside tabu

`\tabu@footnotes` The (global) token that stores the footnotes inside `tabu`:

```
383 \newtoks\tabu@footnotes
```

`\tabu@footnotetext` The macros in case `hyperref` is not used, or used with the option `hyperfootnotes=false`:

```

384 \long\def\tabu@footnotetext #1{%
385   \edef\@tempa{\the\tabu@footnotes
386     \noexpand\footnotetext [\the\csname c@\@mpfn\endcsname]}%
387   \global\tabu@footnotes\expandafter{\@tempa {#1}}}%
388 \long\def\tabu@xfootnotetext [#1]#2{%
389   \global\tabu@footnotes\expandafter{\the\tabu@footnotes
390     \footnotetext [{#1}]{#2}}
391 \let\tabu@xfootnote \@xfootnote

```

`\tabu@Hy@fnttext` The macros in case `hyperref` is loaded with the option `hyperfootnotes=true`:

`\tabu@Hy@xfootnote`

```

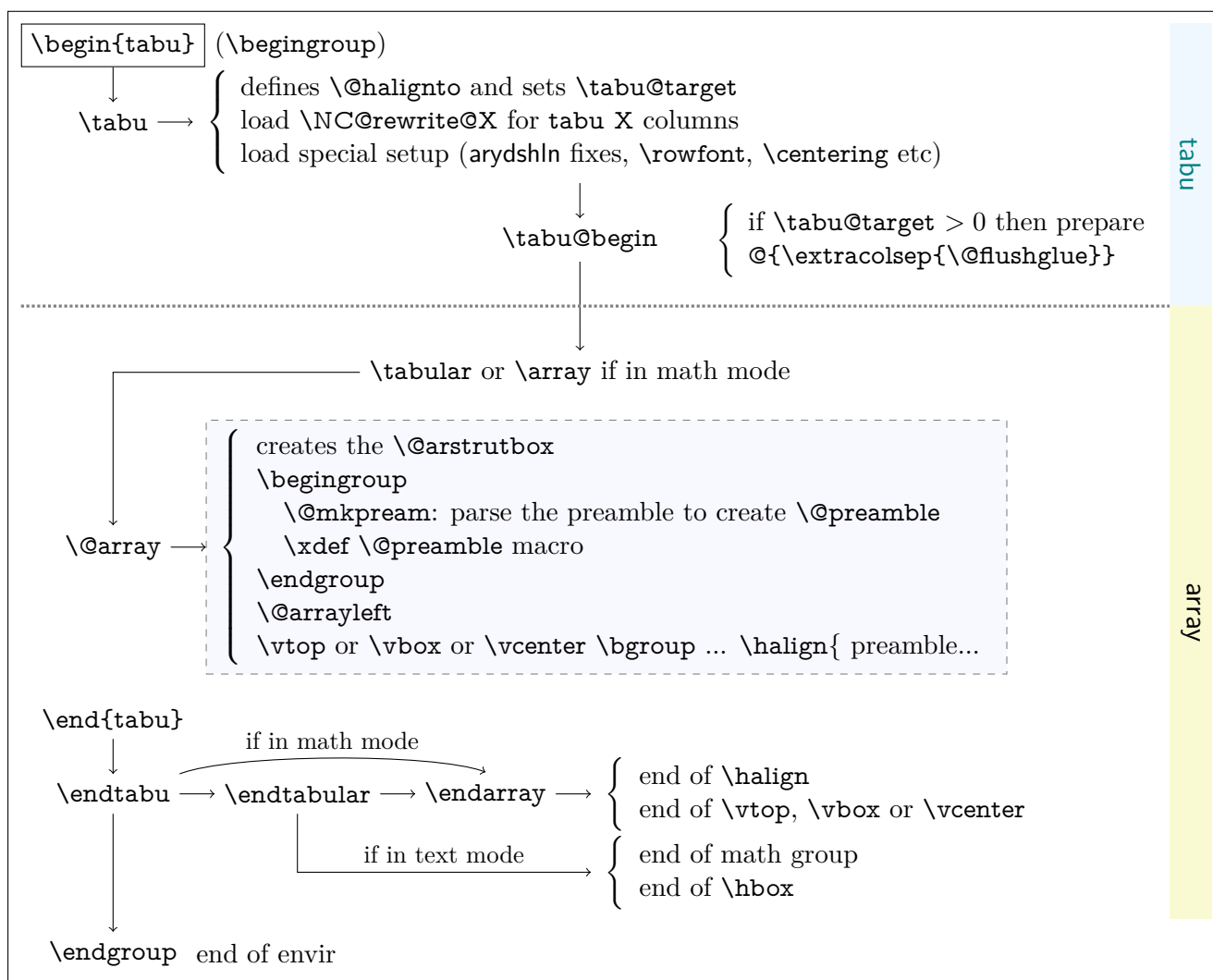
392 \long\def\tabu@Hy@fnttext{\tabu@Hy@fnttxt {\the \c@footnote }}
393 \long\def\tabu@Hy@xfootnote [#1] {%
394   \begingroup
395     \value\@mpfn #1\relax
396     \protected@xdef \@thefnmark {\thempfn}%
397   \endgroup
398   \@footnotemark \tabu@Hy@fnttxt {#1}%
399 }% \tabu@Hy@xfootnote
400 \long\def\tabu@Hy@fnttxt #1#2{%
401   \edef\@tempa{%
402     \the\tabu@footnotes
403     \begingroup
404       \value\@mpfn #1\relax
405       \noexpand\protected@xdef
406         \noexpand\@thefnmark{\noexpand\thempfn}%
407       \expandafter\noexpand\expandafter\tabu@Hy@footnotetext
408       \expandafter{%
409         \Hy@footnote@currentHref
410       }%
411     }%
412     \global\tabu@footnotes\expandafter{%
413       \@tempa {#2}}%
414     \endgroup
415   }%
416 }% \tabu@Hy@fnttxt
417 \long\def\tabu@Hy@footnotetext #1#2{%
418   \H@@footnotetext{%
419     \ifHy@nesting
420       \hyper@@anchor {#1}{#2}%
421     \else
422       \Hy@raisedlink{%
423         \hyper@@anchor {#1}{\relax}%
424       }%
425       \def\@currentHref {#1}%
426       \let\@currentlabelname \@empty
427     #2%

```

```
428     \fi
429   }%
430 }% \tabu@Hy@footnotetext
431 \AtBeginDocument{%
432   \csname\ifcsname ifHy@hyperfootnotes\endcsname
433     ifHy@hyperfootnotes\else iffalse\fi\endcsname
434     \let\tabu@footnotetext \tabu@Hy@ftntext
435     \let\tabu@xfootnote   \tabu@Hy@xfootnote
436   \fi
437 }
```

## 9.5 Flow chart of expansion

General case: `tabu`, `tabu to` and `tabu spread`



### `tabu to` with `X` column

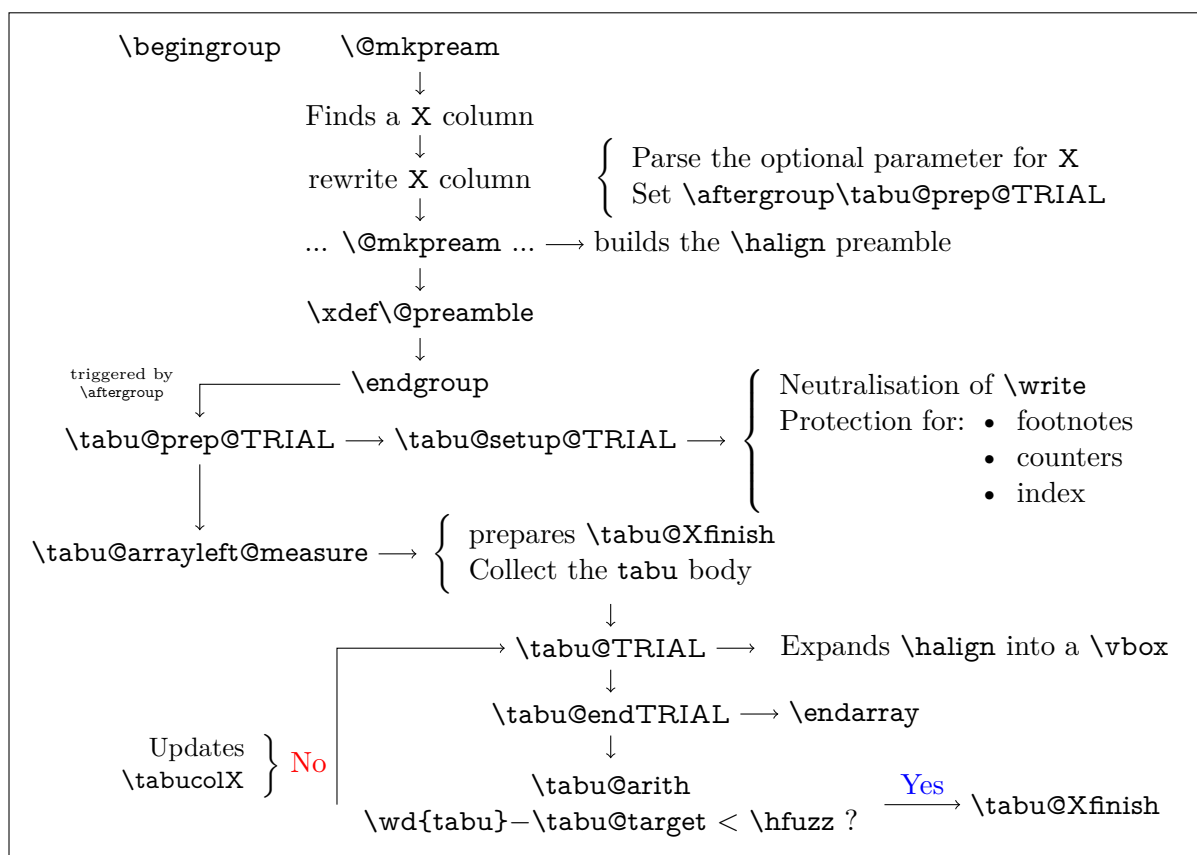
The important part of the job is made inside the dashed box above: `\mkpreamble` expands the columns definitions, which can be user defined. Hopefully, it does its job inside a group, therefore a user-column can set a macro to be expanded `\aftergroup`. This implementation allows much modifications in the tabular preparation, without any change in the macros of `array.sty`.

When a `tabu X` column is found in the preamble by `\mkpreamble`, `tabu` changes his strategy: the macro `\tabu@prep@TRIAL` is set to be expanded `\aftergroup` that is, just after the preamble (`\@preamble`) has been built. This macro does some setup for `tabu` trials to reach the target with variable `X` column widths and gobbles everything until the next `\bgroup` which corresponds to the `\vtop`, `\vbox` or `\vcenter` for the whole tabular. This part of `\@array` is stored into `\tabu@Xfinish` to be expanded after the last trial gave satisfaction to reach the `tabu` target. Then `\tabu@collect` is expanded to find the end of the `tabu` environment, temporarily storing the environment content into a token register.

The last part of `\@array` until `\halign` is expanded inside a `\vbox` which is stored into the box register `\tabu@box` for measuring purpose. `\halign` ends by `\endarray` which stops the `\tabu@box` as well, and then `\tabu@arith` is expanded to compute the gap between the width of `\tabu@box` and the target, and `\tabucolX` (the dimension that correspond to `X[1]`) is updated accordingly.

The trials are “protected” by `{\ifnum0=}\fi` : they occur in a group that will be closed at the very beginning of `\tabu@Xfinish`, when the final tabular will be printed actually. This protection is absolutely necessary to be able to collect the environment body in the case of nested `tabu` with `X` columns. This is related to T<sub>E</sub>X mechanism of expansion inside `\halign` (T<sub>E</sub>X stops

reading when it encounters a & alignment tab character and goes backward expanding anything that were not expanded before).



## tabu spread with X column

In the case of “tabu spread” with X columns, the process is the same as the one described for “tabu to” with X columns. However, the first trial is different because we have first to measure the *natural width* of the tabular. The process is the following:

- `\tabu@target` is first set to `\linewidth` (or `\linegoal` with the `linegoal` package option).
- The X column corresponds to a `\vbox` with `\hsize` fixed to `\tabu@target`.
- Inside this `\vbox` the cell content is written into a `\hbox` whose width is limited to `\tabu@target`. This `\hbox` is captured into the box register `\tabu@box`.
- At the end of the cell, the `\badness` of the `\hbox` is checked:
  - if the `\badness` is  $> 1000$  then the text is too long and “tabu spread” is useless: `tabu to \tabu@target` give the same result.
  - Otherwise, we get the natural width of the cell content by:
 

```
\setbox \tabu@box \hbox {\unhbox \tabu@box}
```
- At the end of the first trial, `\tabu@spreadarith` checks if:

$$\text{width}(\text{tabular}) + \text{spread} < \text{\linewidth (or \linegoal)}$$

- if not, then `tabu to \tabu@target` give the same result
- Otherwise, the target for `tabu to` will be:

$$\text{width}(\text{tabular}) + \text{spread} - \underbrace{\sum_i \text{natural widths } X_i + \text{Max}_i \left( \frac{\text{natural width } X_i}{\text{coef}_i} \right) \times \sum_i \text{coef}_i}_{\text{minimal natural width that can be obtained with the given coefs}}$$

And the next trial will be done as if the user called “tabu to” with this target.



## 9.6 tabu X column definition

`\tabu@rewrite@X` This macro is expanded by `\@mkpream` in case a X column is found.

`\tabu@X@sum` (a dimen) store the sum of the width coefficients. For the first X column found in the preamble, a special setup occurs:

- the default target (either `\linewidth` or `\linegoal` if available) is set if it has not been specified by the user.
- `\@halign` to `\let` to `\relax` to avoid its expansion in `\xdef\@preamble` just after `\@mkpream`. Indeed as long as we have to measure the natural width of the tabular, `\@halign` must be empty for trial steps.
- The rest of the setup is made `\aftergroup` (*ie.* after `\xdef\@preamble` which occurs inside a group) by `\tabu@prep@TRIAL`.

```

438 \def\tabu@rewrite@X #1{\tabu@Xarg {#1}%
439   \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \relax
440   \iftabu@measuring % not the first X column found in preamble
441     \edef\tabu@global@X {\tabu@global@X
442       \def\csname tabu@X\the\tabu@X@cols\endcsname {\strip@pt\tabu@X@sum}%
443       \advance\tabu@X@sum \the\tabu@X@sum\relax}%
444   \else % first X column found in preamble
445     \tabu@measuringtrue
446     \iftabu@spread \tabu@spreadtarget\tabu@target \tabu@target\z@
447     \else          \tabu@spreadtarget \z@ \fi
448     \ifdim\tabu@target=\z@
449       \setlength\tabu@target \tabu@defaulttarget
450       \tabu@message{(tabu) Default target:
451         \ifx\tabu@defaulttarget\linewidth\string\linewidth
452         \else\ifx\tabu@defaulttarget\linegoal\string\linegoal
453         \fi\fi=\the\tabu@target \on@line, page \the\c@page}\fi
454     \tabu@message{\tabu@titles}%
455     \edef\tabu@global@X {\tabu@global@X
456       \def\csname tabu@X1\endcsname{\strip@pt\tabu@X@sum}%
457       \tabu@X@sum \the\tabu@X@sum\relax
458       \tabu@target \the\tabu@target\relax
459       \tabu@spreadtarget \the\tabu@spreadtarget\relax}%
460     \let\tabu@halignto \relax
461   \fi
462 }% \tabu@rewrite@X
463 \let\tabu@@rewrite@X \tabu@rewrite@X

```

`\tabu@rewrite@Xnested` This macro replaces `\tabu@rewrite@X` when tabu makes a trial for X columns.

`\tabu@rewrite@Xrestore` This macro replaces `\tabu@rewrite@X` in the case of `\usetabu`.

```

464 \def\tabu@rewrite@Xnested #1{\in@ ${#1}\ifin@
465   \def\tabucolX@rw {>{${#1}}p{\tabucolX}<{${#1}}}\else
466   \def\tabucolX@rw {p{\tabucolX}}\fi}
467 \def\tabu@rewrite@Xrestore #1{\tabu@gobbleopt{\def\tabucolX@rw{1}}}}

```

`\tabu@Xarg` A tedious (and fastidious) macro to parse the optional argument of X columns. The aim is to built `\tabucolX@rw` which expands to the column specification:

`\tabu@Xparse` `\tabucolX@do` `>{alignment} p or m or b {\dimexpr coef \tabucolX\relax}`

After that array.sty make it easy: `\expandafter\NC@find\tabucolX@rw`

```

468 \def\tabu@Xarg#1{%
469   \advance\tabu@X@cols \@ne
470   \tabu@wdlet\tabu@X@cols \z@
471   \ifx\#1\% <shortcut when no option>
472     \tabu@X@sum \p@

```

```

473 \edef\tabucolX@rw {p{\tabu@dimexpr{\the\tabu@Xcols}{\tabucolX}}}%
474 \edef\tabu@temp {p{\the\tabucolX}}% <required for \usetabu>
475 \else
476 \tabu@Xsum \z@
477 \let\tabucolX@align \@empty \let\tabucolX@rw \@empty
478 \let\tabu@Xmath \@empty \let\tabu@temp \@empty
479 \tabu@Xparse {}#1,\@nnil
480 \fi
481 }% \tabu@Xarg
482 \def\tabu@Xparse #1{\futurelet\@let@token \tabu@Xtest}
483 \expandafter\def\expandafter\tabu@Xparsespace\space{\tabu@Xparse{}}
484 \def\tabu@Xtest{%
485 \ifcase \ifx \@nnil\@let@token \z@ \else
486 \if ,\@let@token \m@ne\else
487 \if p\@let@token 1\else
488 \if m\@let@token 2\else
489 \if b\@let@token 3\else
490 \if l\@let@token 4\else
491 \if c\@let@token 5\else
492 \if r\@let@token 6\else
493 \if j\@let@token 7\else
494 \if L\@let@token 8\else
495 \if C\@let@token 9\else
496 \if R\@let@token 10\else
497 \if J\@let@token 11\else
498 \ifx \@sptoken\@let@token 12\else
499 \if .\@let@token 13\else
500 \if -\@let@token 13\else
501 \ifcat $\@let@token 14\else
502 15\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\relax
503 \tabucolX@do \expandafter \remove@to@nnil
504 \or \tabu@Xtype {p}%
505 \or \tabu@Xtype {m}%
506 \or \tabu@Xtype {b}%
507 \or \tabu@Xalign \raggedright\relax
508 \or \tabu@Xalign \centering\relax
509 \or \tabu@Xalign \raggedleft\relax
510 \or \tabu@Xalign \tabu@justify\relax
511 \or \tabu@Xalign \RaggedRight\raggedright
512 \or \tabu@Xalign \Centering\centering
513 \or \tabu@Xalign \RaggedLeft\raggedleft
514 \or \tabu@Xalign \justifying\tabu@justify
515 \or \expandafter \tabu@Xparsespace
516 \or \expandafter \tabu@Xcoef
517 \or \let\tabu@Xmath=$\expandafter \tabu@Xparse
518 \or \tabu@Xcoef{}%
519 \else\expandafter \tabu@Xparse
520 \fi
521 }% \tabu@Xtest
522 \def\tabu@Xalign#1#2{%
523 \ifx \tabucolX@align\@empty \else \PackageWarning\tabu{
524 {Duplicate horizontal alignment specification}}\fi
525 \ifdefined#1\def\tabucolX@align{#1}\else
526 \def\tabucolX@align{#2}\fi
527 \expandafter\tabu@Xparse
528 }% \tabu@Xalign
529 \def\tabu@Xtype#1{%
530 \ifx \tabucolX@rw\@empty \else \PackageWarning\tabu{

```

```

531     {Duplicate vertical alignment specification}\fi
532     \def\tabucolX@rw{#1}\expandafter\tabu@Xparse
533 }% \tabu@Xtype
534 \def\tabu@Xcoef#1{\edef\tabu@temp{\tabu@temp#1}%
535     \afterassignment\tabu@Xc@ef \tabu@cnt\number\if-#10\fi
536 }% \tabu@Xcoef
537 \def\tabu@Xc@ef{%
538     \advance\tabu@X@sum \tabu@temp\the\tabu@cnt\p@
539     \tabu@Xparse}%
540 }% \tabu@Xc@ef
541 \def\tabucolX@do{%
542     \ifx\tabucolX@rw\@empty \def\tabucolX@rw{p}\fi
543     \ifdim\tabu@X@sum<\z@ \tabu@negcoeftrue
544         \tabu@X@sum \m@ne\tabu@X@sum
545         \tabu@wdlet \tabu@X@cols \p@
546     \else \ifdim \tabu@X@sum=\z@ \tabu@X@sum \p@ \fi
547     \fi
548     \begingroup
549         \toks@{\expandafter {\tabucolX@align}}%
550         \edef\tabu@temp {\strip@pt\tabu@X@sum \tabucolX}%
551         \edef\tabucolX@rw##1{%
552             \ifx$\tabu@Xmath>\{\noexpand\the\toks@ $\}\else
553             \ifx \tabucolX@align\@empty \else
554                 >\{\noexpand\the\toks@\}\fi\fi
555             \tabucolX@rw{##1}%
556             \ifx$\tabu@Xmath<\{$\}\fi}%
557         \edef\x{\endgroup
558             \def\noexpand\tabu@temp {\tabucolX@rw{\the\dimexpr\tabu@temp}}%
559             \def\noexpand\tabucolX@rw{%
560                 \tabucolX@rw{\tabu@dimexpr{\the\tabu@X@cols}{\tabu@temp}}}%
561         }\x
562 }% \tabucolX@do
563 \def\tabu@dimexpr#1#2{%
564     \ifdim \tabu@wd{#1}>\p@
565         \ifdim \tabu@wd{#1}<#2 \tabu@wd{#1}\else#2\fi\else #2\fi
566 }% \tabu@dimexpr

```

## 9.7 Before trials to reach the target: setup after \@mkpream

`\tabu@prep@TRIAL` `\@mkpream` does its job inside a semi-simple group. At the end, `\@preamble` is expanded with `\xdef`, and the group is left: this triggers the expansion of `\tabu@prep@TRIAL` set `\aftergroup` by the first X column encountered in the preamble.

We `\let \@halignto` to `\@empty`: it's a measurement, and put some protections. A group is opened with the famous `\ifnum0='}\fi` and control is given to `\tabu@arrayleft@measure`.

```

567 \def\tabu@prep@TRIAL{%
568     \begingroup \iffalse{\fi \ifnum0='}\fi
569     \tabu@setup@TRIAL
570     \tabucolX \tabu@target
571     \ifcase 0\iftabu@negcoef \else\iftabu@spread \else 1\fi\fi
572     \advance\tabu@X@cols \@ne
573     \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \@undefined
574     \tabu@X@cols \z@
575     \def\tabu@naturalX@max {\z@}%
576     \def\tabu@naturalX@min {\z@}%
577     \let\tabu@startpboxORI \@startpbox
578     \let\@startpbox \tabu@measure@startpbox
579     \fi

```

```

580 \def\tabu@lasttry{\m@ne\p@}\tabu@cnt \z@ % number of trials
581 \tabu@arrayleft@measure
582 }% \tabu@prep@TRIAL

\tabu@setup@TRIAL

583 \def\tabu@setup@TRIAL{\tabu@global@X \let\tabu@global@X \@empty
584 \let\@halignto \@empty \tabu@naturalX \z@
585 \tabu@savecounters \let\tabu@savecounters \relax
586 \hbadness \@M \let\hbadness \@tempcnta
587 \hfuzz \maxdimen \let\hfuzz \@tempdima
588 \let\tabu@rewrite@X \tabu@rewrite@Xnested
589 \let\savetabu \@gobble \let\tabubefaulttarget \linewidth
590 \let\write \tabu@nowrite \let\GenericError \tabu@GenericError
591 \long\def\@xfootnote[##1]##2{}\let\@footnotetext \@gobble
592 \ifx\@arrayright\relax \let\@arrayright \@empty\fi
593 \let\tabu@verticalmeasure \@empty
594 \let\tabu@verticalspacing \@empty
595 \tabu@TRIAL@hook
596 }% \tabu@setup@TRIAL
597 \def\tabu@savecounters{%
598 \def\@elt##1{\global\value{##1}\the\value{##1}\relax}%
599 \edef\tabu@global@X {\cl@ckpt}\let\@elt \relax
600 }% \tabu@savecounters

```

\tabuDisableCommands

```

601 \let\tabu@TRIAL@hook \@empty
602 \newcommand\tabuDisableCommands[1]{\g@addto@macro\tabu@TRIAL@hook {#1}}

```

## 9.8 One trial after another

\tabu@arrayleftmeasure Measuring the whole tabular occurs just before \@arrayleft. Hence the name of the macro: \tabu@arrayleft@measure.

\tabu@TRIAL \halign is temporarily expanded inside a \vbox which is captured in \tabu@box.

```

603 \def\tabu@arrayleftmeasure#1\bggroup{%
604 \def\tabu@Xfinish{%
605 \tabu@global@Xnested \ifnum0='{ }\fi\endgroup
606 \ifdim \abovetabulinesep>\z@ \setbox\@arstrutbox\box\voidb@x
607 \else\ifdim\belowtabulinesep>\z@ \setbox\@arstrutbox\box\voidb@x
608 \fi\fi
609 \tabu@global@X #1\bggroup}%
610 \def\tabu@TRIAL {\setbox\tabu@box \hbox\bgroup $#1\bgroup
611 \the\toks@ \tabu@endTRIAL}%
612 \toks@{\tabu@setup@everycr
613 \let\@preamble \tabu@savedpreamble}% <required for multicolumn>
614 \tabu@collect
615 }% \tabu@arrayleftmeasure

```

\tabu@long@arrayleft@measure This is the long version for longtabu

```

616 \long\def\tabu@long@arrayleft@measure #1\LT@bchunk #2\LT@bchunk{%
617 \def\tabu@Xfinish{%
618 \tabu@global@Xnested \ifnum0='{ }\fi\endgroup
619 \tabu@setup@everycr \let\@preamble \tabu@savedpreamble
620 \tabu@global@X #1\LT@bchunk #2\LT@bchunk}%
621 \def\tabu@TRIAL{%
622 \setbox\tabu@box \hbox\bgroup
623 \tabu@setup@everycr \let\@preamble \tabu@savedpreamble
624 \tabu@global@X \LTchunksizes\@M

```

```

625      #1\LT@bchunk #2\LT@bchunk \the\toks@ \tabu@long@endTRIAL}%
626 \toks@{}%\let\LT@output\relax
627 \tabu@collect
628 }% \tabu@long@arrayleft@measure
629 \def\tabu@long@endTRIAL{%
630   \LT@echunk
631   \global\setbox\@ne \hbox{\unhbox\@ne}\kern\wd\@ne
632   \LT@get@widths
633   \egroup          % got \tabu@box
634   \iftabu@spread \tabu@spreadarith % <only once>
635   \else           \tabu@arith
636   \fi
637   \iftabu@measuring \tabu@measuringfalse
638   \expandafter \tabu@TRIAL % <continue trials>
639   \else
640     \expandafter \tabu@Xfinish
641     \the \expandafter\toks@ \tabu@endenvir % <we are then!>
642   \fi
643 }% \tabu@long@endTRIAL

```

`\tabu@collect` The mechanism is the same as  $\mathcal{M}\mathcal{S}$ -`\collect@body` (also defined in `environ.sty`). The content of the tabular is captured inside `\toks@`, expanded by `\tabu@TRIAL`.

```

644 \def\tabu@collect{\catcode'\^@=13\def\tabu@stack{b}\tabu@collectbody}
645 \long\def\tabu@collectbody#1\end#2{%
646   \edef\tabu@stack{\tabu@pushbegins #1\begin\end\expandafter\@gobble\tabu@stack}%
647   \ifx\tabu@stack\@empty
648     \toks@\expandafter{\the\toks@#1}\def\tabu@endenvir{\end{#2}}%
649     \expandafter \tabu@TRIAL
650   \else
651     \toks@\expandafter{\the\toks@#1\end{#2}}%
652     \expandafter \tabu@collectbody
653   \fi
654 }% \tabu@collectbody
655 \long\def\tabu@pushbegins#1\begin#2{%
656   \ifx\end#2\else b\expandafter\tabu@pushbegins\fi}

```

`\tabu@endTRIAL` At the end of the trial, we call `\tabu@arith` to compute the widths. `\tabu@arith` exits leaving `\iftabu@measuring` equal to `\iftrue`: a further trial is necessary, or equal to `\iffalse`: the target is reached, `\tabu@Xfinish` can print the `tabu` in a last expansion of `\halign`.

```

657 \def\tabu@endTRIAL{%
658   \tabu@adl@endTRIAL % <arydshln in nested trials>
659   \endarray$\egroup
660   \iftabu@spread \tabu@spreadarith % <only once>
661   \else           \tabu@arith
662   \fi
663   \iftabu@measuring \tabu@measuringfalse
664   \expandafter \tabu@TRIAL % <continue trials>
665   \else
666     \expandafter \tabu@Xfinish
667     \the \expandafter\toks@ \tabu@endenvir % <we are then!>
668   \fi
669 }% \tabu@endTRIAL
670 \def\tabu@adl@endTRIAL{\crcr \noalign{\global\adl@ncol \tabu@nbcols}}

```

## 9.9 Algorithms: Measuring the tabu

### The arithmetic of X columns

`\tabu@arith` General algorithms for tabu to with X columns.

```

671 \def\tabu@arith{%
672   \advance\tabu@cnt \@ne \tabu@message{\tabu@message@arith}%
673   \ifx \@startpbox\tabu@measure@startpbox
674     \let\@startpbox \tabu@startpboxORI \fi
675   \dimen@ \tabu@X@sum
676   \iftabu@negcoef \tabu@arith@negcoef \fi
677   \ifcase 0\ifdim \dimexpr\wd\tabu@box-\tabu@hfuzz-1sp<\tabu@target
678     \else \ifdim \tabu@lasttry=\wd\tabu@box 1\else
679     2\fi\fi\relax
680   \ifdim \tabu@X@sum=\dimen@ \tabu@message{\tabu@message@reached}%
681   \else\ifdim\tabu@X@sum=\z@ \tabu@message{\tabu@message@reached}%
682   \else\ifnum \tabu@cnt>\@ne \tabu@message{\tabu@message@reached}%
683   \else \tabu@measuringtrue
684     \tabucolX = \dimexpr\tabucolX*\p@/\tabu@X@sum\relax
685   \fi\fi\fi
686   \or \tabu@message{\tabu@header Reached minimum width. Backing up}%
687   \tabu@warn{\tabu width = \the\wd\tabu@box\space>
688     \the\tabu@target\space+ \tabu@hfuzz}%
689   \or
690     \ifdim \tabu@X@sum=\z@ \tabu@measuringtrue \tabucolX=\@tempdima
691     \else
692       \advance\tabucolX-\dimexpr(\wd\tabu@box-\tabu@target-.5\tabu@X@sum/\p@)
693       *\p@/\tabu@X@sum\relax
694     \ifdim \tabucolX<\z@
695       \tabu@warn{X Columns too narrow (table too wide)}%
696       \tabucolX=\tabucolX@error
697     \else \tabu@measuringtrue
698       \ifdim \tabu@X@sum=\dimen@
699         \edef\tabu@lasttry{\the\wd\tabu@box}%
700       \else
701         \ifnum \tabu@cnt>\@ne \edef\tabu@lasttry{\the\wd\tabu@box}\fi
702         \ifdim \tabucolX<\@tempdima \tabucolX \@tempdima\fi
703       \fi
704     \fi
705   \fi
706   \fi %(ifcase)
707   \iftabu@measuring
708   \else \edef\tabu@global@X{\tabu@global@X
709     \tabucolX \the\tabucolX\relax
710     \tabu@target \the\tabu@target\relax % < for \savetabu >
711     \ifnum \tabu@nested=\z@
712     \ifx \@arrayright\@empty
713     \ifdim \tabu@X@sum>\z@
714       \def\noexpand\@halignto{to\the\tabu@target}\fi\fi\fi}%
715   \fi
716   \tabu@X@sum \dimen@
717 }% \tabu@arith
718 \def\tabu@arith@negcoef{%
719   \tabu@X@cols \@ne \count@ \tabu@X@cols
720   \@whiles \ifcsname tabu@X\the\tabu@X@cols\endcsname\fi{%
721     \ifdim \tabu@wd\tabu@X@cols>\p@
722     \expandafter\let\expandafter\tabu@temp
723     \csname tabu@X\the\tabu@X@cols\endcsname

```



```

724         \ifdim \tabu@wd\tabu@X@cols <\tabu@temp\tabucolX
725             \advance\tabu@X@sum -\tabu@temp\p@
726             \tabu@message{\tabu@message@negcoef{\the\tabu@X@cols}}}%
727             \ifdim \tabu@wd\tabu@X@cols >\tabu@wd\count@ \count@ \tabu@X@cols \fi
728         \fi\fi
729         \advance\tabu@X@cols \@ne
730     }%
731     \@tempdima=\dimexpr\tabu@wd\count@ * \p@/
732             \dimexpr\csname tabu@X\the\count@\endcsname\p@\relax\relax
733 }% \tabu@arith@negcoef

```

### The arithmetic of X columns for tabu spread

```
\tabu@spreadarith Algorithm for tabu spread with X columns: only the first pass.
```

```

734 \def\tabu@spreadarith{%
735   \tabu@naturalX \tabu@naturalX@max\relax
736   \tabucolX \tabu@naturalX@min\relax
737   \tabu@message{\tabu@message@spread}%
738   \ifdim \dimexpr\wd\tabu@box + \tabu@spreadtarget <\tabu@target
739     \advance\tabu@spreadtarget
740     \dimexpr\wd\tabu@box-\tabu@naturalX+\tabucolX\relax
741     \ifdim \tabu@spreadtarget <\tabu@target
742       \tabu@target =\tabu@spreadtarget
743     \fi
744     \tabu@message{\the\tabu@target}%
745     \tabu@message{\ifdim \tabu@spreadtarget>\tabu@target
746       (tabu) spread reduced to fit default target.\fi}%
747   \else
748     \tabu@message{\expandafter\tabu@msg@align \the\tabu@target { }{}{}{}{}\@@}
749     \tabu@message{(tabu)\ifdim\wd\tabu@box<\tabu@target spread too large\else
750       tabu spread is useless here\fi,
751       default target used.}%
752   \fi
753   \let\@startpbox \tabu@startpboxORI
754   \xdef\tabu@halignto {to\the\tabu@target}%
755   \tabucolX \tabu@target
756   \tabu@spreadfalse \tabu@measuringtrue
757 }% \tabu@spreadarith

```

### Reporting in the .log file (debugshow option)

```

\tabu@message@arith
\tabu@message@spread
\tabu@message@negcoef
258 \def\tabu@message@arith{\tabu@header
259   \expandafter\tabu@msg@align \the\tabucolX { }{ }{ }{ }\@@
260   \expandafter\tabu@msg@align \the\wd\tabu@box { }{ }{ }{ }\@@
261   \expandafter\tabu@msg@align \the\tabu@target { }{ }{ }{ }\@@
262   \expandafter\tabu@msg@align \the\tabu@X@sum {}{}{}{}{}\@@
263   \ifx\@startpbox\tabu@measure@startpbox\tabu@spaces (coefs<0)\else
264   \ifdim\tabu@X@sum>\z@
265     \expandafter\tabu@msg@align \the\dimexpr(\tabu@target-\wd\tabu@box)
266                                     *\p@/\tabu@X@sum\relax {}{}{}{}{}\@@\fi\fi^^J}%
267 \def\tabu@message@spread{\tabu@spreadheader
268   \expandafter\tabu@msg@align \the\tabu@spreadtarget {}{}{}{}{}\@@
269   \expandafter\tabu@msg@align \the\wd\tabu@box {}{}{}{}{}\@@
270   \expandafter\tabu@msg@align \the\tabucolX {}{}{}{}{}\@@
271   \expandafter\tabu@msg@align \expandafter-\the\tabu@naturalX {}{}{}{}{}\@@}
272 \def\tabu@message@reached{\tabu@header
273   \expandafter\tabu@msg@align \the\tabucolX { }{ }{ }{ }\@@

```



```

774 \expandafter\tabu@msg@align \the\wd\tabu@box {}{}{}{}\@@
775 <\expandafter\tabu@msg@align \the\tabu@target { }{ }{ }{}\@@
776 + \tabu@hfuzz\space= hfuzz *****~J%
777 (tabu)\tabu@spaces***** Reached Target:
778     hfuzz = \tabu@hfuzz\on@line\space *****~J%
779     \ifnum\tabu@nested=\z@ \tabu@elapsedtime ~^Jfi}%
780 \def\tabu@message@etime{(tabu)\tabu@spaces
781   Time elapsed for trials:
782   \the\numexpr(\pdfelapsedtime-\tabu@starttime-32767)/65536\relax sec
783   \the\numexpr\numexpr(\pdfelapsedtime-\tabu@starttime)
784     -\numexpr(\pdfelapsedtime-\tabu@starttime-32767)/65536\relax*65536\relax
785       *1000/65536\relax ms}%
786 \def\tabu@message@negcoef #1{\tabu@spaces\tabu@spaces
787   * #1. X[-\csname tabuX#1\endcsname]: natural width
788   = \tabu@wd{#1}
789   < \csname tabuX#1\endcsname X
790   = \the\dimexpr\csname tabuX#1\endcsname\tabucolX
791 }% \tabu@message@negcoef
792 \edef\tabu@spaces{\@spaces}
793 \def\do#1{%
794   \def\tabu@msg@align##1.##2##3##4##5##6##7##8##9\@@{%
795     \ifnum##1<10 #1 #1\else
796     \ifnum##1<100 #1 \else
797     \ifnum##1<\@m #1\fi\fi\fi
798     ##1.##2##3##4##5##6##7##8##1}%
799   \def\tabu@header{(tabu) \ifnum\tabu@cmt<10 #1\fi\the\tabu@cmt) #1}%
800   \def\tabu@titles{\ifnum \tabu@nested=\z@
801     (tabu) Try#1 #1 tabu X #1 #1 #1tabu Width #1 #1 Target
802         #1 #1 #1 Coefs #1 #1 #1 Update~^Jfi}%
803   \def\tabu@spreadheader{%
804     (tabu) Try#1 #1 Spread #1 #1 tabu Width #1 #1 Nat. Min. #1 #1 #1 Nat.X
805                                     #1 #1 #1 New Target~^J%
806     (tabu)spread}%
807   \def\tabu@tracing@save##1{\begingroup
808     \def\colsep #####1\relax      {(tabu) column sep #1 #1 #1 = #####1~^J}%
809     \def|arrayrulewidth #####1\relax{(tabu) arrayrulewidth #1 = #####1~^J}%
810     \def\doublerulesep #####1\relax {(tabu) doublerulesep #1 #1= #####1~^J}%
811     \def\extrarowheight #####1\relax{(tabu) extrarowheight #1 = #####1~^J}%
812     \def\extrarowdepth #####1\relax {(tabu) extrarowdepth #1 #1= #####1~^J}%
813     \def\extratabsurround #####1\relax{(tabu) extratabsurround = #####1~^J}%
814     \def\minrowclearance #####1\relax {(tabu) minrowclearance#1 = #####1~^J}%
815     \def\arraystretch #####1{(tabu) arraystretch #1 #1 = #####1~^J}%
816     \iftabu@colortbl
817     \def\CT@arc#####1{(tabu) arrayrulecolor #1 = #####1~^J}%
818     \def\CT@drcs#####1{(tabu) doublerulesepcolor=#####1~^J}%
819     \let\color \relax
820     \fi
821     \def\NC@find #####1\@nil{(tabu) tabu preamble#1 #1 = \detokenize{#####1}~^J}%
822     \def\tabu@wdddef#####1####2{(tabu) Natural width #####1 = #####2~^J}%
823     \let\edef \@gobbletwo \let\def \@empty
824     \tabu@message{%
825       (tabu) \string\savetabu{##1}: \on@line~^J%
826     \ifx \savetabu\tabusaveX
827       (tabu) X columns width #1 = \the\tabucolX~^J\fi
828       (tabu) target #1 #1 #1 #1 #1 = \the\tabu@target~^J%
829       \tabu@sav@param
830       \tabu@sav@pream \@nil
831       \iftabu@negcoef \tabu@savewd\fi}%

```

```

832     \endgroup}%
833 }\do{ }

```

## Measuring the natural width for tabu spread and negativ width-coefficients

`\tabu@measure@startpbox` When “tabu spread” is used with X columns, the first trial must measure the natural width of the columns. When X columns have negativ coefficient, the natural is computed after the target has been reached, with the absolute coefficients.

Nested trials may occur (tabu spread inside a X column with negativ coefficient for example).

For the furthur trials, the standard scheme for X column is used: the natural width is measured only once.

pdfT<sub>E</sub>X font expansion is disabled inside the `varwidth` environment (we set `\pdfadjustspacing` to 0).

```

834 \def\tabu@measure@startpbox#1{\bgroup
835   \let\@startpbox \tabu@startpboxORI           % in case of nesting
836   \in@\tabu@dimexpr{#1}%
837   \ifin@
838     \global\advance\tabu@X@cols \@ne
839     \ifcsname tabu@X\the\tabu@X@cols\endcsname\else
840       \global\tabu@X@cols \@ne
841       \global\tabu@naturalX \z@
842     \fi
843   \fi
844   \iftabu@spread\else
845     \ifin@ {\let\tabu@dimexpr \@firstoftwo\expandafter}%
846     \ifdim \tabu@wd{#1}<\p@ \in@false\fi\fi\fi
847   \ifin@
848     \begingroup \let\tabu@dimexpr \@secondoftwo
849     \edef\x{\endgroup \def\noexpand\tabu@temp{\the\dimexpr#1}%
850     }\x
851     \setbox\tabu@box=\hbox\bgroup
852     \let\tabu@rewrite@X \tabu@@rewrite@X
853     \tabu@nocellspace
854     \begin{varwidth}{\tabu@temp}\narrowragged\arraybackslash
855       \tabu@noadjustspacing \parfillskip\@flushglue
856       \bgroup \aftergroup\tabu@measure@endpbox
857       \ifdefined \cellspacetoplimit \bgroup\color@begingroup\fi
858   \else
859     \expandafter\@gobble\@startpbox{#1}% \@gobble \bgroup
860   \fi
861 }% \tabu@measure@startpbox
862 \def\tabu@nocellspace{\def\bcolumn##1\@nil{\let\ecolumn\@empty}
863 \ifdefined\pdfadjustspacing
864   \def\tabu@noadjustspacing{\pdfadjustspacing\z@}%
865 \else
866   \let\tabu@noadjustspacing\relax
867 \fi
868 \def\tabu@measure@endpbox{%
869   \@finalstrut\@arstrutbox\end{varwidth}\egroup      % <got my \tabu@box>
870   \global\advance\tabu@naturalX \wd\tabu@box
871   \ifdim \tabu@naturalX@max<\tabu@naturalX
872     \xdef\tabu@naturalX@max{\the\tabu@naturalX}\fi
873   \@tempdima=\dimexpr \wd\tabu@box * \tabu@X@sum /
874     \dimexpr\csname tabu@X\the\tabu@X@cols\endcsname\p@\relax\relax
875   \ifdim \tabu@naturalX@min<\@tempdima
876     \xdef\tabu@naturalX@min{\the\@tempdima}\fi

```

```

877 \ifdim \tabu@wd\tabu@X@cols>\z@
878 \ifdim \tabu@wd\tabu@X@cols<\wd\tabu@box
879 \tabu@wddef\tabu@X@cols {\the\wd\tabu@box}%
880 \tabu@debug{\tabu@spaces\tabu@spaces
881 * \the\tabu@X@cols. X[-\csname tabu@X\the\tabu@X@cols\endcsname]:
882 target=\the\dimexpr\tabu@temp\relax\space
883 \expandafter\expandafter\expandafter\string\tabu@wd\tabu@X@cols
884 =\tabu@wd\tabu@X@cols}%
885 \fi\fi
886 \box\tabu@box \egroup % end of \vtop (measure)
887 }% \tabu@measure@endpbox
888 \def\tabu@wd#1{\csname tabu@W\the\tabu@nested @\number#1\endcsname}
889 \def\tabu@wddef#1{\expandafter\xdef
890 \csname tabu@W\the\tabu@nested @\number#1\endcsname}
891 \def\tabu@wdlet#1{\global\expandafter\let
892 \csname tabu@W\the\tabu@nested @\number#1\endcsname}
893 \def\tabu@naturalX@min{\z@}
894 \def\tabu@naturalX@max{\z@}

```

## 9.10 \everycr setup and \tabuphantomline

`\tabu@setup@everycr` `\ialign` resets `\everycr` to an empty token. This macro sets `\everycr` for the `tabu` environment.

```

895 \def\tabu@setup@everycr{%
896 \tabu@firstcline>true
897 \let\tabu@first@ccline \z@
898 \def\tabu@restoreeverycr {\let\everycr \tabu@everycr}%
899 \def\everycr{\afterassignment\tabu@restoreeverycr \@temptokena}% <for ialign>
900 \tabu@everycr\expandafter{\the\tabu@everycr
901 \noalign{%
902 \ifcase\tabu@first@ccline
903 \global\let\tabu@first@ccline \tw@
904 \or\global\let\tabu@first@ccline \tw@
905 \or\global\tabu@firstclinefalse
906 \global\let\tabu@first@ccline \m@ne
907 \fi}%
908 \tabu@rowfontreset
909 }%
910 }% \tabu@setup@everycr

```

`\tabuphantomline` This macro inserts a phantom line in front of a `tabu`. This is necessary when you use `\usetabu` with `tabu X` column, with a single line containing `\multicolumn...`

```

911 \def\tabuphantomline{\crrc\noalign{%
912 \global\let\@arstrutbox \voidb@x
913 \iftabu@colortbl
914 \xdef\tabu@minrowclearance{\the\minrowclearance}%
915 \global\minrowclearance \z@
916 \fi
917 \global\let\vccenter \vbox
918 \ifnum \tabu@nested>\z@ \vspace*{\dimexpr\extrarowheight-\ht\@arstrutbox}\fi
919 \toks@{}\count@\@ne
920 \@whilenum\count@<\tabu@nbcols\do{\advance\count@\@ne
921 \toks@\expandafter{\the\toks@&}}%
922 \toks@\expandafter{\the\toks@
923 \cr\noalign{%
924 \global\let\@arstrutbox \tabu@arstrutbox
925 \global\let\vccenter \tabu@vccenter
926 \iftabu@colortbl

```

```

927         \global\minrowclearance \tabu@minrowclearance\relax\fi
928         \global\let\tabu@first@cline \one}}}%
929     \expandafter\the\toks@
930 }% \tabuphantomline

```

## 9.11 Lines inside tabu

### Vertical lines

`\tabuvline@rewrite` | is defined as a new column type but only inside the group where `\@mkpream` does its job. This way, the modification is very local to `tabu`.

`\tabu@lines` This feature is disabled with the `light` option.

```

931 \def\tabu@lines{{\endlinechar\mone \scantokens{\def\x{}}}%
932   \expandafter\expandafter\tabu@lines@ \x
933 }% \tabu@lines
934 \def\tabu@lines@#1{%
935   \tabu@newcolumnntype #1{\tabuvline@rewrite}%
936   \NC@list\expandafter{\the\NC@list \NC@do #1}%
937 }% \tabu@lines@
938 \newcommand*\tabuvline@rewrite[1][\]{\tabuvline@arg{#1}%
939   \expandafter \NC@find \tabu@temp}
940 \def\tabu@vline#1{\vrule width#1}

```

`\tabuvline@arg` A tedious (and fastidious) macro to parse the optional argument of | vertical lines...

```

941 \def\tabuvline@arg#1{%
942   \ifx\#1\% <shortcut when no option>
943     \edef\tabu@temp{!}%
944     \iftabu@colortbl\beginngroup\noexpand\CT@arc@ \fi
945     \noexpand\tabu@vline\arrayrulewidth
946     \iftabu@colortbl\endgroup\fi}}%
947   \else
948     \ifcsname tabu@line@style@\detokenize {#1}\endcsname
949       \def\tabu@temp##1\relax##2\relax{##1}%
950       \expandafter\tabu@temp\romannumeral-'\q
951       \csname tabu@line@style@\detokenize {#1}\endcsname
952       \tabuvline@argiii
953     \else \futurelet \tabu@temp \tabuvline@argi #1\p@\p@\@nnil \tabuvline@argiii
954     \fi
955   \fi
956 }% \tabuvline@arg
957 \def\tabuvline@argi{%
958   \let\tabu@color \@empty
959   \ifcat A\noexpand\tabu@temp
960     \@tempdima\arrayrulewidth
961     \expandafter\tabu@getlinecolor
962   \else \expandafter\tabuvline@argii
963   \fi
964 }% \tabuvline@argi
965 \def\tabuvline@argii#1\@nnil{%
966   \tabu@maybecolor \@tempdima #1 \@nnil
967 }% \tabuvline@argii
968 \def\tabuvline@argiii{%
969   \edef\tabu@temp{!\beginngroup
970     \ifx\tabu@color\@empty
971       \iftabu@colortbl \noexpand\CT@arc@ \fi
972     \else \tabu@color\fi
973     \noexpand\tabu@vline{\the\@tempdima}\endgroup}}%

```

```
974 }% \tabuvline@argiii
```

## Horizontal lines: \tabucline

`\tabucline` `\tabucline[{style or spec.}]{start-end}`

`\tabucline` appears only at the end of a line: this is the place where we can insert a `\noalign` group. We built a new line to be inserted inside the `tabu`: this new line which contains the rule or leaders, is stored into `\toks@` (inside the `\noalign` group). When leaving the group `\toks@` is expanded.

```
975 \def\tabu@linedash{4pt}%    <default value>
976 \def\tabu@dashgap{4pt}%    <default value>
977 \def\tabucline{\noalign{\ifnum0='}\fi \ifstar
978   {\@tempswatrue\tabu@ccline}
979   {\@tempswafalse\tabu@ccline}}%
980 \newcommand*\tabu@ccline[2][\toks@]{%
981   \tabu@startstop{#2}\tabu@getlinespec{#1}\@multicnt\@ne
982   \ifnum\tabu@start>\tabu@stop
983   \else\ifx\tabu@xleaders\relax
984   \else
985     \toks@{\noalign{\global\let\@arstrutbox \voidb@x}}%
986   %%      \iftabu@firstccline\if\tabu@align t\vskip-\ht\@arstrutbox\fi\fi
987     \@whilenum\@multicnt<\tabu@start\do{\advance\@multicnt\@ne
988       \toks@\expandafter{\the\toks@ &}}%
989     \loop
990       \toks@\expandafter{\the\toks@ \omit }%
991       \ifnum \@multicnt=\@ne
992       \expandafter \tabu@add \csname tabu@tok@\the\@multicnt L\endcsname \@nil
993       \fi
994       \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
995         \toks@\expandafter{\the\toks@ \begingroup}%
996         \toks@\expandafter{\the\expandafter\toks@
997           \tabu@xleaders \endgroup}%
998       \fi
999       \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
1000       \expandafter \tabu@add \csname tabu@tok@\the\@multicnt R\endcsname \@nil
1001       \fi
1002       \ifnum\@multicnt<\tabu@stop
1003         \advance\@multicnt\@ne
1004         \if@tempswa \toks@\expandafter{\the\toks@ &}%
1005         \else      \toks@\expandafter{\the\toks@ \span}\fi
1006       \repeat
1007       \@whilenum\@multicnt<\tabu@nbc\do{\advance\@multicnt\@ne
1008         \toks@\expandafter{\the\toks@ &}}%
1009       \toks@\expandafter{\the\toks@ \cr
1010         \noalign{\global\let\@arstrutbox \tabu@arstrutbox}}%
1011   %%      \tabu@debug{LIGNE AJOUTEE:\the\toks@}%
1012   \fi\fi
1013   \ifnum0='{ \fi\expandafter}\the\toks@
1014 }% \tabu@ccline
1015 \def\tabu@add #1\@nil{\toks@\expandafter{\the\toks@
1016   \begingroup #1\endgroup\ifmmode$\fi}}
```

`\tabu@startstop` This macro parses the mandatory argument of `\tabucline`: start-column and end-column of the cline.

```
1017 \def\tabu@startstop#1{\tabu@start@stop #1\relax 1-\tabu@nbc\@nnil}
1018 \def\tabu@start@stop #1-#2\@nnil{%
1019   \@defaultunits \tabu@start \number0#1\relax\@nnil
1020   \@defaultunits \tabu@stop \number0#2\relax\@nnil
```

```

1021 \ifnum\tabu@start>\tabu@nbc\cols\tabu@start \tabu@nbc\cols\else
1022 \ifnum\tabu@start<\@ne \tabu@start \@ne \fi\fi
1023 \ifnum\tabu@stop>\tabu@nbc\cols \tabu@stop \tabu@nbc\cols \else
1024 \ifnum\tabu@stop<\@ne \tabu@stop \tabu@nbc\cols\fi\fi
1025 }% \tabu@start@stop

```

`\tabu@getlinespec` This macro parses the optional argument of `\tabucline` and check if it's a line specification (then `\tabu@getline` is expanded) or a `\leaders` specification (then `\tabu@leaders` is expanded).

```

1026 \def\tabu@getlinespec#1{\let\tabu@xleaders \relax \let\tabu@color \@empty
1027 \@defaultunits \let\@tempa=#1 \relax\@nnil
1028 \ifx\@tempa\relax \let\tabu@xleaders \tabu@defaultleaders\else
1029 \ifx\@tempa\hbox \tabu@defleaders{#1}\else
1030 \ifx\@tempa\box \tabu@defleaders{#1}\else
1031 \ifx\@tempa\copy \tabu@defleaders{#1}\else
1032 \ifcsname tabu@line@style@\detokenize{#1}\endcsname
1033 \csname tabu@line@style@\detokenize{#1}\endcsname
1034 \else \tabu@getline{#1}\p@ on0pt off0pt\fi\fi\fi\fi\fi
1035 }% \tabu@getlinespec
1036 \def\tabu@defleaders#1{%
1037 \def\tabu@xleaders{\iftabu@colortbl\CT@arc@\fi
1038 \xleaders\hbox{\lower.5\extrarowheight#1}\tabu@leaderfill}}
1039 \def\tabucline@warn#1{\PackageWarning\tabu{
1040 {Undefined line style: #1
1041 \MessageBreak Using default line style instead}}%
1042 \let\tabu@xleaders \tabu@defaultleaders
1043 }% \tabucline@warn

```

`\tabu@getline` This macro parses the optional argument of `\tabucline` (or the one of `\tabulinestyle`) and extract the thickness, the dash and gap specified. Default values assignments are done either.

`\tabu@maybecolor`

```

1044 \def\tabu@getline#1{\tabu@lineon #1 \@nil on\tabu@linedash \p@ \@nil\@nnil{#1}}
1045 \def\tabu@maybecolor{\ifx\tabu@color\@empty \afterassignment\tabu@colortest
1046 \else \@defaultunits\fi}
1047 \def\tabu@lineon #1on#2\@nil#3\@nnil#4{%
1048 \@defaultunits \let\@tempa=#1 \relax\@nnil
1049 \ifcase 0\ifx o\@tempa 1\else\ifcat A\@tempa 2\fi\fi\relax
1050 \tabu@maybecolor \@tempdima #1\arrayrulewidth \p@\@nnil
1051 \tabu@maybecolor \@tempdimb #2\p@ \@nnil
1052 \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil
1053 \or\@tempdima \arrayrulewidth
1054 \tabu@maybecolor \@tempdimb #2\p@ \@nnil
1055 \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil
1056 \else
1057 \tabu@maybecolor \@tempdima \arrayrulewidth #1\p@\@nnil
1058 \ifx\tabu@color\@empty \tabucline@warn\@tempa\else
1059 \@tempdimb \z@
1060 \tabu@lineoff off0pt \p@\@nnil \fi
1061 \fi
1062 }% \tabu@lineon
1063 \def\tabu@lineoff #1off#2\@nnil{%
1064 \tabu@maybecolor \@tempdimc #2\p@ \@nnil
1065 \iftabu@colortbl \ifx \tabu@color\@empty
1066 \def\tabu@color{\noexpand\CT@arc{}}%
1067 \fi\fi
1068 \ifdim \@tempdimb=\z@
1069 \ifdim \@tempdimc>\z@ \@tempdimb \tabu@dashgap\relax\fi\fi
1070 \ifdim \@tempdimc=\z@

```



```

1071 \ifdim \@tempdimb>\z@ \@tempdimc \tabu@linedash\relax\fi\fi
1072 \ifdim \@tempdima<\z@ \else
1073 \ifdim \@tempdimb<\z@ \else
1074 \ifdim \@tempdimc<\z@ \else
1075 \edef \tabu@xleaders{\tabu@color\xleaders
1076 \ifdim \@tempdimc>\z@
1077 \hbox\bgroup \kern\the\dimexpr \@tempdimc/2\relax\fi
1078 \noexpand\iftabu@firstcline
1079 \vrule depth\dimexpr\the\@tempdima
1080 \ifdim \@tempdimb>\z@ width\the\@tempdimb\fi
1081 \noexpand\else
1082 \vrule height\dimexpr-\extrarowheight+\the\@tempdima
1083 depth\dimexpr\extrarowheight
1084 \ifdim \@tempdimb>\z@ width\the\@tempdimb\fi
1085 \noexpand\fi
1086 \ifdim \@tempdimc>\z@
1087 \kern\the\dimexpr \@tempdimc/2\egroup\fi
1088 \tabu@leaderfill}%
1089 \fi\fi\fi
1090 }% \tabu@lineoff

```

### \tabu@colortest

```

1091 \def\tabu@colortest{\futurelet\tabu@temp\tabu@linecolor}
1092 \def\tabu@linecolor{%
1093 \ifcase 0\if ,\noexpand\tabu@temp\else
1094 \ifx\relax\tabu@temp\else
1095 \ifx \@sptoken\tabu@temp1\else
1096 \ifcat A\noexpand\tabu@temp2\else
1097 3\fi\fi\fi\fi\relax
1098 \def\tabu@next##1{\futurelet\tabu@temp\tabu@linecolor}%
1099 \or\def\tabu@next{\tabu@gobblespace{\futurelet\tabu@temp\tabu@linecolor}}%
1100 \or\let\tabu@next \tabu@getlinecolor
1101 \else\expandafter\remove@to@nnil
1102 \fi \tabu@next
1103 }% \tabu@linecolor
1104 \def\tabu@getlinecolor#1\p@{%
1105 \edef\@tempa{\zap@space #1 \@empty}%
1106 \ifcsname\string\color@\@tempa\endcsname
1107 \edef\tabu@color{\noexpand\noexpand\noexpand\color{\@tempa}}% \set@color
1108 \fi\remove@to@nnil
1109 }% \tabu@getlinecolor

```

### \tabulinestyle \tabulinestyle{style=spec.}

```

1110 \def\tabulinestyle#1{\@for\@tempa:=#1\do
1111 \expandafter\tabu@linestyle\@tempa==\@nil}}
1112 \def\tabu@linestyle#1=#2=#3\@nil{%
1113 \begingroup \tabu@getlinespec {#2}%
1114 \toks@\expandafter{\tabu@xleaders}%
1115 \@temptokena\expandafter{\tabu@color}%
1116 \expandafter\xdef\csname tabu@line@style@\detokenize{#1}\endcsname
1117 {\@tempdima\the\@tempdima
1118 \def\noexpand\tabu@color{\the\@temptokena}\relax
1119 \def\noexpand\tabu@xleaders{\the\toks@}\relax}%
1120 \endgroup
1121 }% \tabu@linestyle
1122 \expandafter\def\csname tabu@line@style@\endcsname {%
1123 \let\tabu@xleaders \tabu@defaultleaders}%
1124 \def\tabu@defaultleaders{\iftabu@colortbl\CT@arc@\fi

```



```

1125 \leaders
1126 \iftabu@firstcline
1127 \vrule depth \arrayrulewidth
1128 \else
1129 \vrule height\dimexpr-\extrarowheight+\arrayrulewidth
1130 depth \extrarowheight
1131 \fi
1132 \tabu@leaderfill}
1133 \let\tabu@leaderfill \hfil

```

## 9.12 Verbatim inside tabu with X columns

`\tabu@sanitizetext`

```

1134 {\catcode32=13\relax\catcode'\^@=13\relax
1135 \gdef\tabu@verb{\@sanitize\catcode'\^=7\edef\string~}\tabu@makeatletter
1136 \catcode32=13\let =\ \catcode'\^@=13\def^^@{\hskip\parfillskip\null\linebreak}%
1137 \endlinechar\m@ne}}
1138 {\catcode64=13\edef @{\string @}\expandafter\xdef
1139 \csname tabu@makeatletter\endcsname{\catcode64=13\def\noexpand @{\unskip @}}
1140 \newcommand\tabu@sanitizetext[1][\ttfamily]{\begingroup
1141 \tabu@verb #1\tabu@s@nitizetext}
1142 \long\def\tabu@s@nitizetext#1{\@makeother\{\@makeother\}%
1143 \everyeof{\noexpand}\scantokens{#1}\endgroup}

```

## 9.13 Numbers in tabu

`\tabudecimal`

`\tabudecimal` `\tabu@tabudecimal` is `\tabudecimal` inside `tabu`.

```

1144 \def\tabu@tabudecimal#1{%
1145 \def\tabu@decimal{#1}\@temptokena{}%
1146 \let\tabu@getdecimal@ \tabu@getdecimal@ignorespaces
1147 \tabu@scandecimal
1148 }% \tabu@tabudecimal
1149 \def\tabu@scandecimal{\futurelet \tabu@temp \tabu@getdecimal@}
1150 \def\tabu@skipdecimal#1{#1\tabu@scandecimal}
1151 \def\tabu@getdecimal@ignorespaces{%
1152 \ifcase 0\ifx\tabu@temp\ignorespaces\else
1153 \ifx\tabu@temp\@sptoken1\else
1154 2\fi\fi\relax
1155 \let\tabu@getdecimal@ \tabu@getdecimal
1156 \expandafter\tabu@skipdecimal
1157 \or \expandafter\tabu@gobblespace\expandafter\tabu@scandecimal
1158 \else \expandafter\tabu@skipdecimal
1159 \fi
1160 }% \tabu@getdecimal@ignorespaces
1161 \def\tabu@get@decimal#1{\@temptokena\expandafter{\the\@temptokena #1}%
1162 \tabu@scandecimal}
1163 \def\do#1{%
1164 \def\tabu@get@decimalspace#1{%
1165 \@temptokena\expandafter{\the\@temptokena #1}\tabu@scandecimal}%
1166 }\do{ }
1167 \let\tabu@@tabudecimal \tabu@tabudecimal

```

`\tabu@getdecimal`

```

1168 \def\tabu@getdecimal{%
1169 \ifcase 0\ifx 0\tabu@temp\else
1170 \ifx 1\tabu@temp\else

```

### 9.14 \extrarowsep and \tabulinesep

The macro may be prefixed by `\global`.

The macro may be prefixed by `\global`.

48 / 67

```

1219 \belowtabulinesep\abovetabulinesep
1220 \tabu@Gsave \tabu@G@linesep \tabu@C@linesep \abovetabulinesep \belowtabulinesep
1221 }% \tabu@setlinesep
1222 \def\tabu@Glinesep{\ifx \tabu@G@linesep\@empty \else \tabu@Rlinesep\fi}
1223 \def\tabu@Rlinesep{\tabu@Grestore \tabu@G@linesep \tabu@C@linesep}
1224 \let\tabu@C@linesep \z@
1225 \let\tabu@G@linesep \@empty

```

`\tabu@Gsave` Utility macro to implement the possibility to prefix a macro by `\global`.

```

\tabu@Grestore 1226 \def\tabu@Gsave #1#2#3#4{\xdef#1{#1%
1227 \toks#2{\toks\the\currentgrouplevel{\global#3\the#3\global#4\the#4}}}%
1228 }% \tabu@Gsave
1229 \def\tabu@Grestore#1#2{\begingroup
1230 \toks#2{\#1\toks\currentgrouplevel\expandafter{\expandafter}\the\toks#2\relax
1231 \ifcat$\the\toks\currentgrouplevel$\else
1232 \global\let#1\@empty \global\let#2\z@
1233 \fi
1234 \expandafter\endgroup
1235 \the\toks\currentgrouplevel
1236 }% \tabu@Grestore

```

`\tabu@verticalmeasure`

```

1237 \def\tabu@verticalmeasure{%
1238 \ifnum \currentgrouptype>12
1239 \begingroup\lrbox\tabu@box \dollarbegin
1240 \let\tabu@verticalspacing \tabu@verticalsp@lcr
1241 \else
1242 \setbox\tabu@box
1243 \ifnum \currentgrouptype=5\vtop\else\vbox\fi \bgroup
1244 \everypar{}\let\@finalstrut \@gobble
1245 \let\tabu@verticalspacing \tabu@verticalsp@pmb
1246 \fi
1247 }% \tabu@verticalmeasure

```

`\tabu@verticalsp@lcr` Vertical spacing adjustment for standard l, c, r columns.

```

1248 \def\tabu@verticalsp@lcr{% adjust \ht \tabu@box and \dp \tabu@box
1249 \dollarend\endlrbox\endgroup
1250 \@tempdima \dimexpr \ht\@arstrutbox -\ht\tabu@box \relax
1251 \@tempdimb \dimexpr \dp\@arstrutbox -\dp\tabu@box \relax
1252 \ifdim \@tempdima<\abovetabulinesep
1253 \ht\tabu@box=\dimexpr \ht\tabu@box+\abovetabulinesep\relax \fi
1254 \ifdim \@tempdimb<\belowtabulinesep
1255 \dp\tabu@box=\dimexpr \dp\tabu@box+\belowtabulinesep \relax \fi
1256 \leavevmode \box\tabu@box
1257 }% \tabu@verticalsp@lcr

```

`\tabu@verticalsp@pmb` Vertical spacing adjustment with struts for p, m, or b columns.

```

1258 \def\tabu@verticalsp@pmb{% inserts struts as needed
1259 \par \expandafter\egroup
1260 \expandafter\@tempdimc \the\prevdepth
1261 \@tempdima \dimexpr \ht\@arstrutbox -\ht\tabu@box \relax
1262 \@tempdimb \dimexpr \dp\@arstrutbox -\@tempdimc \relax
1263 \ifdim \@tempdima<\abovetabulinesep
1264 \llap {\vrule
1265 \@height \dimexpr\ht\tabu@box+\abovetabulinesep\relax
1266 \@depth -\ht\tabu@box
1267 \@width \tabustrutrule
1268 }\nointerlineskip \vskip-\ht\tabu@box

```

```

1269 \fi
1270 \unvbox\tabu@box
1271 \ifdim \@tempdimb<\belowtabulinesep
1272 \llap{\vrule
1273 \@depth \dimexpr \belowtabulinesep \relax
1274 \@height \z@
1275 \@width \tabustrutrule
1276 }\nointerlineskip%\vskip-\@tempdimc
1277 \let\@finalstrut \@gobble
1278 \fi
1279 }% \tabu@verticalsp@pmb

```

## 9.15 \savetabu

**\savetabu** When this command is called by the user, the tabu preamble and target are globally stored into a macro `\tabu@saved@{user-name}`.

```

1280 \newcommand*\savetabu[1]{\noalign{%
1281 \ifx\@#1\\ \tabu@savewarn{}\else
1282 \ifundefined\tabu@saved@{\string#1}{\tabu@savewarn{#1}{Overwriting}}%
1283 {%
1284 \toks0\expandafter{\tabu@savedpreamble}%
1285 \toks1\expandafter{\tabu@savedparam}%
1286 \toks2\expandafter{\tabu@savedpream}%
1287 \toks3{}}%
1288 \iftabu@negcoef
1289 \edef\tabu@savewd{\tabu@savewd{\number\numexpr1}}%
1290 \toks3\expandafter{\tabu@savewd}\fi
1291 \expandafter\xdef\csname tabu@saved@\string#1\endcsname##1{%
1292 \noexpand\ifcase##1\relax
1293 \noexpand\ifdim\tabu@target>\z@
1294 \noexpand\tabu@warn@usetabu
1295 \noexpand\fi
1296 \tabu@target \the\tabu@target\relax
1297 \tabucolX \the\tabucolX\relax
1298 \the\toks1%
1299 \gdef\noexpand\tabu@usetabu{%
1300 \the\toks1%
1301 \def\noexpand\@halignto{\tabu@halignto}%
1302 \tabucolX \the\tabucolX\relax
1303 \def\noexpand\@preamble{\the\toks0}%
1304 \noexpand\tabu@make@arstrutbox
1305 \the\toks3
1306 \noexpand\ifx \noexpand\tabu@align\noexpand\tabu@aligndefault@text
1307 \def\noexpand\tabu@align{\tabu@align}%
1308 \noexpand\fi}%
1309 \aftergroup\noexpand\tabu@usetabu
1310 \noexpand\else
1311 \noexpand\ifdim\tabu@target=\z@
1312 \tabu@target \the\tabu@target\relax
1313 \noexpand\fi
1314 \gdef\noexpand\tabu@preamble{%
1315 \noexpand\ifx \noexpand\tabu@align\noexpand\tabu@aligndefault@text
1316 \def\noexpand\tabu@align{\tabu@align}%
1317 \noexpand\fi}%
1318 \aftergroup\noexpand\tabu@preamble
1319 \noexpand\fi
1320 \the\toks2}%
1321 \tabu@message@save{#1}}%

```

```

1322 \fi}%
1323 }% \savetabu
1324 \def\tabu@aligndefault@text{\tabu@aligndefault}%
1325 \def\tabu@warn@usetabu{\PackageWarning{tabu}
1326   {Specifying a target with \string\usetabu\space is useless
1327   \MessageBreak The target cannot be changed!}}
1328 \def\tabu@savewd#1{%
1329   \ifnum#1>\tabu@nbc\cols \expandafter \@gobble
1330   \else \ifdim\tabu@wd{#1}>\z@
1331     \noexpand\tabu@wddef{#1}{\tabu@wd{#1}}\fi
1332     \expandafter\tabu@savewd % loop
1333   \fi{#1+1}}%
1334 }% \tabu@savewd
1335 \def\tabu@make@arstrutbox{%
1336   \@tempdima \ht \strutbox
1337   \@tempdimb \dp \strutbox
1338   \advance \@tempdima by\extrarowheight
1339   \advance \@tempdimb by\extrarowdepth
1340   \setbox \@arstrutbox \hbox{\vrule
1341     \@height \arraystretch \@tempdima
1342     \@depth \arraystretch \@tempdimb
1343     \@width \z@}%
1344 }% \tabu@make@arstrutbox

```

`\tabu@savewarn` Info for overwriting when `\savetabu` is used.

`\tabu@saveerr` Error if `\usetabu` is called with an unknown argument.

```

1345 \def\tabu@savewarn#1#2{\PackageInfo{tabu}
1346   {User-name ‘#1’ already used for \string\savetabu
1347   \MessageBreak #2}}%
1348 \def\tabu@saveerr#1{\PackageError{tabu}
1349   {User-name ‘#1’ is unknown for \string\usetabu
1350   \MessageBreak I cannot restore an unknown preamble!}\@ehd}

```

## 9.16 `\rowfont`

### Setting font and alignment specification

`\rowfont` `\rowfont` uses the control sequences `\tabu@celllalign`, `\tabu@cellleft`, `\tabu@cellright` and `\tabu@cellralign` which have been placed on purpose into the user-defined tokens inserted in any preamble by the `array` package.

`\tabu@celllalign` and `\tabu@cellralign` are used to modify the alignment. If the optional [*alignment*] parameter of `\rowfont` is not specified, then those control sequence expand to `\@empty`.

`\tabu@cellleft` contains the font-modification information.

Placement of those control sequences into the user-tokens that are inserted in the preamble by the `array` package is explained below under the macro `\tabu@prepnext@tok`.

```

1351 \def\tabu@rowfont{\noalign{\ifnum0='}\fi\tabu@row@font}
1352 \newcommand*\tabu@row@font[2][]{%
1353   \global\let\tabu@rowfontreset \tabu@rowfont@reset
1354   \global\let\tabu@cellleft \tabu@cellleft
1355   \global\let\tabu@cellright \tabu@cellright
1356   \global\let\tabu@celllalign \tabu@celllalign
1357   \global\let\tabu@cellralign \tabu@cellralign
1358   \gdef\tabu@cellfont{\let\color \tabu@colorleavevmode#2\let\color \tabu@colorORI}%
1359   \ifcsname tabu@cell@#1\endcsname % row alignment
1360     \csname tabu@cell@#1\endcsname \fi

```

```

1361 \toks@{\expandafter {\tabu@cellleft\tabu@cellfont}}% inside \noalign group ok
1362 \xdef\tabu@cellleft {\the\toks@}%
1363 \ifnum0='{ \fi}% end of noalign group
1364 }% \rowfont
1365 \def\tabu@colorleavevmode{\leavevmode\tabu@colorORI}
1366 \AtBeginDocument{\let\tabu@colorORI \color}

```

`\tabu@rowfont@reset` This macro resets `\tabu@celllalign`, `\tabu@cellleft`, `\tabu@cellright`, `\tabu@cellralign` and `\everycr` to the value they had before the expansion of `\rowfont`.

It expands when a new row is inserted into the tabular or array.

```

1367 \def\tabu@rowfont@reset{%
1368   \noalign{%
1369     \global\let\tabu@rowfontreset \@empty
1370     \global\let\tabu@cellleft \tabu@@cellleft
1371     \global\let\tabu@cellright \tabu@@cellright
1372     \global\let\tabu@cellfont \@empty
1373     \global\let\tabu@celllalign \tabu@@celllalign
1374     \global\let\tabu@cellralign \tabu@@cellralign
1375   }%
1376 }% \tabu@rowfont@reset
1377 \let\tabu@rowfontreset \@empty

```

## Preparing stuff to be able to use `\rowfont`

`\tabu@prepnext@tok` `\tabu@prepnext@tok` will replace `\prepnext@tok` (in `array.sty`): its purpose is to add the control sequences `\tabu@celllalign`, `\tabu@cellleft`, `\tabu@cellright` and `\tabu@cellralign` at the right position in the “preamble” for `\halign`. Those control sequences are not inserted directly into the preamble, but by the means of the user-tokens placed there by the `array` package.

The package `array` defines a macro `\prenext@tok` to initialize each user-token inserted at both side of each “normal” column. For “special” `@` and `!` columns, there is only one token.

$$\underbrace{>\{\backslash\bfseries\color{red}\}}_{\backslash\text{toks}<i>} \quad \text{r} \quad \underbrace{<\{\backslash\color{black}\backslash,\backslash\$}\}_{\backslash\text{toks}<i+1>}$$

When a column is inserted in the tabular preamble (`\@preamble`), the T<sub>E</sub>X counter `\count@` is equal to  $i + 1$  (*ie.* the right token) and the counter `\@tempcnta` is equal to  $i$  (*ie.* the left token). If the column is special (*ie.* `@` or `!`) `\@tempcnta` is not updated.

Thus, when a new token is “prepared” by `\prepnext@tok`:

**either:**  $i = \text{count@} = \text{@tempcnta}$  : the token to prepare (*ie.* `\toks<i+1>`) is the right one of a “normal” column. The switch `\iftabu@cellright` is set to `true`.

The *previous* token (`\toks<i> = \toks\count@`) is necessarily the left one of this “normal” column: we prepend `\tabu@celllalign` and append `\tabu@cellleft` to this token (`\toks<i>`). This token is finished and will not change afterwards.

**or:**  $i = \text{count@} = \text{@tempcnta} + 1$  : the token to prepare (`\toks<i+1>`) is either the left one of a normal column, or the single one of a special `@` or `!` column.

If the switch `\iftabu@cellright` is `true`, then the *previous* token `\toks<i>` is the right one of the last inserted column (which was a “normal” column, thus);, `\tabu@cellright\tabu@cellralign` is appended to it, and the switch `\ittabu@cellright` is reset to `false`. May be `\prepnext@tok` will be expanded again (by `\save@decl`): if it happens, then again  $\text{count@} = \text{@tempcnta} + 1$  (same case) but `\iftabu@cellright` is `false` and nothing is changed.

**else:** The token to prepare (which is `\toks<i+1> = \toks\count@ + 1`), cannot be the right one of a “normal” column: `\iftabu@cellright` is set to `false`.

The fact that  $|\text{count@} - \text{@tempcnta}| > 1$  tells us that the previous token `\toks<i>` is necessarily the single one of a “special” `@` or `!` column. We don’t modify this token, as long as *special columns are always inserted as is*: `\rowcolor` has no effect on special columns, nor `\rowfont`.



Thereafter, the original initialisation sequence occurs: `\advance\count@ by\@ne` and initialize the token to prepare (`\toks\count@=\toks<i+1>`) to an empty one.

```

1378 \newif\iftabu@cellright
1379 \AtBeginDocument{\let\tabu@prepnext@tokORI \prepnext@tok}% original definition
1380 \def\tabu@prepnext@tok{%
1381   \ifnum \count@<\z@ % <first initialisation>
1382     \@tempcnta \@M % <not initialized by array.sty>
1383     \tabu@nbcolls\z@
1384     \let\tabu@fornoopORI \fornoop
1385     \expandafter\gdef\expandafter\tabu@global@temp\expandafter{%
1386       \tabu@global@temp\gdef\tabu@global@temp{}\tabu@nbcolls\z@}%
1387     \expandafter\let\csname tabu@tok@1L\endcsname \relax
1388     \tabu@cellrightfalse
1389   \else
1390     \ifcase \numexpr \count@-\@tempcnta \relax % (case 0): prev. token is left
1391       \advance \tabu@nbcolls\@ne
1392       \expandafter\let\csname tabu@tok@\the\tabu@nbcolls R\endcsname \relax
1393       \expandafter\gdef\expandafter\tabu@global@temp\expandafter{%
1394         \tabu@global@temp \advance\tabu@nbcolls\@ne}%
1395       \iftabu@cellright % before-previous token is right and is finished
1396         \tabu@cellrightfalse % <only once>
1397         \tabu@savetok R\tabu@preptokenright
1398       \fi
1399       \ifnum \tabu@nbcolls=\@ne \tabu@savetok L\fi
1400       \tabu@preptokenleft
1401     \or % (case 1) previous token is right
1402       \tabu@savetok R\tabu@cellrighttrue
1403       \let\@fornoop \tabu@lastnoop
1404     \else % special column: do not change the token
1405       \ifnum \tabu@nbcolls>\z@ %special column: always on the right of normal one
1406         \tabu@savetok R%
1407       \else % unless this is the very first column (\tabu@nbcolls=0)
1408         \advance\tabu@nbcolls\@ne \tabu@savetok L\advance\tabu@nbcolls\m@ne
1409       \fi
1410       \iftabu@cellright % before-previous token is right
1411         \tabu@cellrightfalse
1412         \tabu@preptokenright
1413       \fi
1414     \fi % \ifcase
1415   \fi
1416   \tabu@prepnext@tokORI
1417 }% \tabu@prepnext@tok
1418 \long\def\tabu@lastnoop#1\@#2#3{\in@\@nextchar{#2}}%
1419 \ifin@ \let\@fornoop \tabu@fornoopORI \expandafter\prepnext@tok \fi}
1420 \def\tabu@preptokenright{%
1421   \advance \count@ \m@ne
1422   \toks\count@\expandafter {\the\toks\count@ \tabu@cellright \tabu@celllalign}%
1423   \advance \count@ \@ne
1424 }% \tabu@preptokenright
1425 \def\tabu@preptokenleft{\toks\count@\expandafter{\expandafter\tabu@celllalign
1426   \the\toks\count@ \tabu@cellleft}%
1427 }% \tabu@preptokenleft
1428 \def\tabu@savetok#1{\begingroup
1429   \expandafter\tabu@savet@k\csname tabu@tok@\the\tabu@nbcolls #1\endcsname
1430 }% \tabu@savetok
1431 \def\tabu@savet@k#1{%
1432   \ifnum \@chclass=\@ne \toks@\toks\count@

```



```

1433 \else\ifnum\@chclass=5 \toks@\toks\count@
1434 \else \toks@{}\fi\fi
1435 \unless\ifx#1\relax
1436 \toks@\expandafter\expandafter\expandafter{%
1437 \expandafter#1\the\toks@}\fi
1438 \toks@\expandafter{\expandafter\def\expandafter#1\expandafter{%
1439 \the\toks@}}%
1440 \@temptokena\expandafter{\tabu@global@temp}%
1441 \xdef\tabu@global@temp {\the\@temptokena \the\toks@}%
1442 \expandafter\endgroup \the\toks@
1443 }% \tabu@savetok

```

## Neutralisation of glues and alignment modification

```

\tabu@cellleft First initialisation to \@empty.
\tabu@celllalign 1444 \let\tabu@cellleft \@empty
\tabu@cellright 1445 \let\tabu@cellright \@empty
\tabu@cellralign 1446 \def\tabu@celllalign {\tabu@cellleft}% row font spec. applies to pre-column material
\tabu@cellfont 1447 \let\tabu@cellralign \@empty
1448 \let\tabu@cellfont \@empty

\tabu@cell@align
1449 \def\tabu@cell@align #1#2#3{%
1450 \toks@\expandafter{\tabu@celllalign #1}%
1451 \xdef\tabu@celllalign{\the\toks@}%
1452 \toks@\expandafter{\tabu@cellralign #2}%
1453 \xdef\tabu@cellralign{\the\toks@}%
1454 \toks@\expandafter{\tabu@cellleft #3}%
1455 \xdef\tabu@cellleft{\the\toks@}%
1456 }% \tabu@cell@align

\tabu@cell@l Setup macros to modify the alignment. The skips inserted to make the standard alignment
\tabu@cell@c specified in the tabular preamble are not the same with standard array tabulars and colortbl
\tabu@cell@r tabulars, hence the switch \iftabu@colortbl.
\tabu@cell@j 1457 \def\tabu@cell@l{% force alignment to left
1458 \tabu@cell@align
1459 {\tabu@removehfil \raggedright \tabu@cellleft}% left
1460 {\tabu@flush1\tabu@ignorehfil}% right
1461 \raggedright
1462 }% \tabu@cell@l
1463 \def\tabu@cell@c{% force alignment to center
1464 \tabu@cell@align
1465 {\tabu@removehfil \centering \tabu@flush{.5}\tabu@cellleft}
1466 {\tabu@flush{.5}\tabu@ignorehfil}
1467 \centering
1468 }% \tabu@cell@c
1469 \def\tabu@cell@r{% force alignment to right
1470 \tabu@cell@align
1471 {\tabu@removehfil \raggedleft \tabu@flush1\tabu@cellleft}
1472 \tabu@ignorehfil
1473 \raggedleft
1474 }% \tabu@cell@r
1475 \def\tabu@cell@j{% force justification (for p, m, b columns)
1476 \tabu@cell@align
1477 {\tabu@justify\tabu@cellleft}
1478 {}
1479 \tabu@justify
1480 }% \tabu@cell@j

```

```

1481 \def\tabu@justify{%
1482     \leftskip\z@skip \@rightskip\leftskip \rightskip\@rightskip
1483     \parfillskip\@flushglue
1484 }% \tabu@justify
1485 %% ragged2e settings
1486 \def\tabu@cell@L{% force alignment to left (ragged2e)
1487     \tabu@cell@align
1488     {\tabu@removehfil \RaggedRight \arraybackslash
1489         \tabu@cellleft}
1490     {\tabu@flush1\tabu@ignorehfil}
1491     {\RaggedRight \arraybackslash}}%
1492 }% \tabu@cell@L
1493 \def\tabu@cell@C{% force alignment to center (ragged2e)
1494     \tabu@cell@align
1495     {\tabu@removehfil \Centering \arraybackslash
1496         \tabu@flush{.5}\tabu@cellleft}
1497     {\tabu@flush{.5}\tabu@ignorehfil}
1498     {\Centering \arraybackslash}}%
1499 }% \tabu@cell@C
1500 \def\tabu@cell@R{% force alignment to right (ragged2e)
1501     \tabu@cell@align
1502     {\tabu@removehfil \RaggedLeft \arraybackslash
1503         \tabu@flush1\tabu@cellleft}
1504     {\tabu@ignorehfil}
1505     {\RaggedLeft \arraybackslash}}%
1506 }% \tabu@cell@R
1507 \def\tabu@cell@J{% force justification (ragged2e)
1508     \tabu@cell@align
1509     {\justifying \arraybackslash \tabu@cellleft}
1510     {}
1511     {\justifying \arraybackslash}}%
1512 }% \tabu@cell@J
1513 \def\tabu@flush#1{%
1514     \iftabu@colortbl          % colortbl uses \hfill rather than \hfil
1515         \hskip \ifnum\currentgrouptype>13 \stretch{#1}%
1516         \else\ifdim#1pt<1pt \tabu@cellskip
1517         \else \stretch{#1}
1518         \fi\fi \relax
1519     \else                    % array.sty
1520         \ifnum \currentgrouptype>13\relax
1521             \hfil \hskip1sp
1522         \fi
1523     \fi
1524 }% \tabu@flush
1525 \AtBeginDocument{%
1526     \@ifpackageloaded{ragged2e}
1527     {}
1528     {\let\tabu@cell@L \tabu@cell@l
1529         \let\tabu@cell@R \tabu@cell@r
1530         \let\tabu@cell@C \tabu@cell@c
1531         \let\tabu@cell@J \tabu@cell@j
1532     }%
1533 }% AtBeginDocument

```

`\tabu@removehfil` `\tabu@removehfil` removes (eventually) the infinite stretchable glue inserted *before* the cell (in the preamble of `\halign`) to make the column alignment.

```

1534 \newskip\tabu@cellskip
1535 \let\tabu@hfil\hfil
1536 \let\tabu@hfill\hfill
1537 \let\tabu@hskip\hskip
1538 \def\tabu@removehfil{%
1539   \iftabu@colortbl
1540     \unkern \tabu@cellskip = \lastskip
1541     \ifnum\gluestretchorder\tabu@cellskip = \tw@ \hskip-\tabu@cellskip
1542     \else \tabu@cellskip = \z@skip
1543     \fi
1544   \else
1545     \ifdim\lastskip=1sp\unskip\fi
1546     \ifnum\gluestretchorder\lastskip = \@ne
1547       \hfilneg % \hfilneg for array.sty but not for colortbl...
1548     \fi
1549   \fi
1550 }% \tabu@removehfil

```

`\tabu@ignorehfil` `\tabu@ignorehfil` removes (eventually) the infinite stretchable glue inserted *after* the cell (in the preamble of `\halign`) to make the column alignment.

```

1551 \def\tabu@ignorehfil{%
1552   \aftergroup\tabu@nohfil
1553 }% \tabu@ignorehfil
1554 \def\tabu@nohfil{% \hfil -> do nothing + restore original \hfil
1555   \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1556 }% \tabu@nohfil
1557 \AtBeginDocument{%
1558   \@ifpackageloaded{colortbl}
1559   {%
1560     \def\tabu@nohfil{%
1561       \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1562       \def\hfill{\let\hfill\tabu@hfill}% (colortbl uses \hfill) pfff...
1563       \def\hskip##1\relax{\let\hskip\tabu@hskip}}% local
1564     }% \@ifpackageloaded colortbl
1565   }%
1566 }% AtBeginDocument

```

## 9.17 Utilities

### tabu `\fbox`

`\tabu@fbox` works exactly like L<sup>A</sup>T<sub>E</sub>X `\fbox` but allows the syntax: `\fbox\bgroup...\egroup` suitable for use inside tabular columns. `\fbox` is `\let` to `\tabu@fbox` at the entry inside a `tabu` environment.

```

1567 \def\tabu@fbox{%
1568   \leavevmode
1569   \let\color@bgroup\bgroup
1570   \def\color@egroup{\endgraf\egroup}%
1571   \afterassignment\tabu@begin@fbox
1572   \setbox\@tempboxa \hbox
1573 }% \tabu@fbox
1574 \def\tabu@begin@fbox{\color@bgroup\kern\fboxsep\aftergroup\tabu@end@fbox}
1575 \def\tabu@end@fbox{\kern\fboxsep \color@egroup \@frameb@x\relax}

```

**tabu \fcolorbox**

`\tabu@fcolorbox` works exactly like `xcolor \fcolorbox` but allows the syntax:

`\fcolorbox{frame color}{background color}\bgroup...\egroup`

suitable for use inside tabular columns. `\fcolorbox` is `\let` to `\tabu@fcolorbox` at the entry inside a `tabu` environment.

```

1576 \def\tabu@color@b@x#1#2{\leavevmode
1577   \def\tabu@do@color@b@x{#1{#2\color@block{\wd\z@}{\ht\z@}{\dp\z@}\box\z@}}%
1578   \afterassignment\tabu@begin@color@b@x
1579               \setbox\z@ \hbox
1580 }% \tabu@color@b@x
1581 \def\tabu@begin@color@b@x{%
1582   \bgroup \kern\fboxsep \set@color
1583   \aftergroup\tabu@end@color@b@x
1584 }% \tabu@begin@color@b@x
1585 \def\tabu@end@color@b@x{\kern\fboxsep \egroup
1586   \dimen@ \ht\z@ \advance\dimen@\fboxsep \ht\z@ \dimen@
1587   \dimen@ \dp\z@ \advance\dimen@\fboxsep \dp\z@ \dimen@
1588   \tabu@do@color@b@x
1589 }% \tabu@end@color@b@x

```

**\centering, \raggedright, \raggedleft**

Inside `tabu` environment, no need to add `\arraybackslash` after these commands.

```

1590 \def\tabu@temp#1#2{{\toks@ \expandafter{#2\arraybackslash}\xdef#1{\the\toks@}}}
1591 \tabu@temp \tabu@centering \centering
1592 \tabu@temp \tabu@raggedleft \raggedleft
1593 \tabu@temp \tabu@raggedright \raggedright
1594 \def\tabu@trivlist{\expandafter\def\expandafter\@trivlist\expandafter{%
1595   \expandafter\let\expandafter\\ \expandafter\@normalcr\@trivlist}}%

```

## 9.18 Corrections

**delarray compatibility fix for colortbl and arydshln**

Both `colortbl` and `arydshln` forgot the control sequence `\@arrayright` which must be expanded by `\endarray`. Originally defined for `delarray`, this control sequence is used by `tabu` environments when `tabu X` columns are present in the preamble.

Here is the fix. We test if `\endarray` contains `\@arrayright` before modifying the control sequence, in case `colortbl` and/or `arydshln` modify their implementation.

```

1596 \def\tabu@fix@arrayright{%
1597   \@ifpackageloaded{arydshln}
1598   {%
1599     \@ifpackageloaded{colortbl}
1600     {% colortbl + arydshln
1601       \def\tabu@endarray{%
1602         \adl@endarray \egroup \adl@arrayrestore \CT@end \egroup %<original>
1603         \@arrayright % <FC>
1604         \gdef\@preamble{}% <FC>
1605         }% \endarray
1606       {% arydshln / no colortbl
1607         \def\tabu@endarray{%
1608           \adl@endarray \egroup \adl@arrayrestore \egroup %<original>
1609           \@arrayright % <FC>
1610           \gdef\@preamble{}% <FC>
1611           }% \endarray

```

```

1612 }%
1613 {%
1614 \@ifpackageloaded{colortbl}
1615   {% colortbl / no arydshln
1616   \def\tabu@endarray{%
1617     \crrc \egroup \egroup
1618     \@arrayright % <FC>
1619     \gdef\@preamble{}\CT@end
1620   }%
1621   {\PackageWarning{tabu}
1622     {\string\@arrayright\space is missing from the
1623     \MessageBreak definition of \string@endarray.
1624     \MessageBreak Comptability with delarray.sty is broken.}}%
1625 }%
1626 }% \tabu@fix@arrayright

```

### arydshln @ columns

```

1627 \def\tabu@adl@xarraydashrule#1#2#3{%
1628   \ifnum\@lastchclass=\adl@class@start\else
1629   \ifnum\@lastchclass=\@ne\else
1630   \ifnum\@lastchclass=5 \else % <FC> @-arg (class 5) and !-arg (class 1)
1631     \adl@leftrulefalse \fi\fi % must be treated the same
1632   \fi
1633   \ifadl@zwvrule\else \ifadl@inactive\else
1634     \@addtopreamble{\vrule\@width\arrayrulewidth
1635       \@height\z@ \@depth\z@}\fi \fi
1636   \ifadl@leftrule
1637     \@addtopreamble{\adl@vlineL{\CT@arc@}{\adl@dashgapcolor}%
1638       {\number#1}#3}%
1639   \else \adl@vlineR{\CT@arc@}{\adl@dashgapcolor}%
1640     {\number#2}#3}
1641   \fi
1642 }% \tabu@adl@xarraydashrule

```

### arydshln, colors without colortbl and empty p columns

arydshln redefines \@endpbox for p columns. The definition is stored in \adl@act@endpbox. Here it is:

```

\unskip \ifhmode \nobreak
  \vrule\@width\z@\@height\z@\@depth\dp\@arstrutbox
\fi
\egroup \adl@colhtdp \box\adl@box \hfil

```

The \vrule inserted is exactly what package array calls: \@finalstrut \@arstrutbox.

However, just like in array.sty, this array-strut should be inserted inconditionnally, and \ifhmode applies only to \nobreak (misplaced \fi in arydshln definition).

Finally, arydshln is not compatible with colors in columns, such that: >{\color{red}}p3in, Unless colortbl is also loaded, the color group is missing.

Fixed inside tabu.

```


1643 \def\tabu@adl@act@endpbox{%
1644   \unskip \ifhmode \nobreak \fi \adl@finalstrut \@arstrutbox
1645   \egroup \egroup
1646   \adl@colhtdp \box\adl@box \hfil
1647 }% \tabu@adl@act@endpbox
1648 \def\tabu@adl@fix{%
1649   \let\adl@xarraydashrule \tabu@adl@xarraydashrule % <fix> arydshln
1650   \let\adl@act@endpbox \tabu@adl@act@endpbox % <fix> arydshln

```

```
1651 \let\adl@act@@endpbox \tabu@adl@act@endpbox % <fix> arydshln
1652 \let\@preamerror \@preamerr % <fix> arydshln
1653 }% \tabu@adl@fix

1654 \</package>
```

## 10 References

- [1] *A new implementation of L<sup>A</sup>T<sub>E</sub>X's tabular and array environments* by Frank Mittelbach  
2008/09/09 v2.4c – Tabular extension package (FMi)  
[CTAN:help/Catalogue/entries/array.html](http://CTAN:help/Catalogue/entries/array.html)
- [2] *The varwidth package* by Donald Arseneau  
2009/03/30 ver 0.92 – Variable-width minipages  
[CTAN:help/Catalogue/entries/varwidth.html](http://CTAN:help/Catalogue/entries/varwidth.html)
- [3] *The enumitem-zref package* by   
2010/11/28 ver 1.1 – Extended references for enumitem pkg  
[CTAN:help/Catalogue/entries/enumitem-zref.html](http://CTAN:help/Catalogue/entries/enumitem-zref.html)

## 11 History

[2011/01/25 v2.0]

- Vertical spacing of lines implemented ! See `\tabulinesep` and `\extrarowsep`.
- `\tabulinestyle`: user defined line style can now be used inside the optional argument of the `[[...]]` preamble token.
- `[[...]]` is now allowed in `\multicolumn` preamble inside `tabu` environment. (Disabled with the `light` package option.)
- Bug fixed inside `\tabu@prepnexttok` (again !!! - a difficult case !)
- Incompatibility of package `cellspace` with `tabu spread` and `tabu` with `negativ coefficients` for `X` columns with has been lifted.  
However, as said in the documentation of package `cellspace`, `S` column modifier does not work in the case of nested tabulars.  
The `S` column modifier becomes `C` when the package `siunitx` is loaded (see `siunitx` documentation).  
Moreover, `cellspace` does not work with `color` or `xcolor` and paragraph column types !!  
Finally, `cellspace` redefines `\@startpbox` and `\@endpbox` and is therefore not fully compatible with `array.sty` and therefore with `tabu`.  
For all those reasons, `tabu` displays a warning to discourage the use of `cellspace` with the `tabu` environment.

[2011/01/15 v1.9]

- Bug in `\savetabu` when used inside `longtabu...`
- Bug when `tabu` with `X` column is nested inside `lontabu`.
- Documentation (`\rowfont` was missing in the `summary`).

[2010/12/28 v1.8]

- `\tracingtabu` / `debugshow` package option:  
reporting of the time elapsed during trials (if `\pdfelapsedtime` and thus pdf<sub>T</sub>E<sub>X</sub> is available)  
Slight modifications for better reporting on the `.log` file.
- Fix a bug when `\savetabu` is used after `\multicolumn` (`\multicolumn` globally redefines `\@preamble`).
- Fix a bug with `\tabucline` and `\CT@arc@` (`colortbl`).
- Better privacy of columns types specifically defined for `tabu`.
- Improvement in the rewriting process (but only very few people should notice...)
- Documentation.



**[2010/12/18 v1.7]**

- Code optimisation
- Modification in the columns rewriting process (bug with some new column types defined by the user).

**[2010/12/07 v1.5]**

- Implementation of negativ width coefficients for **X** columns (cf. [tabu X columns point 2](#)).
- Columns natural widths computation (for **tabu spread** with **X** columns and negativ coefficients) is based on the code of the **varwidth** package by Donald Arseneau.
- **longtabu** is now provided, based on the **longtable** package by David Carlisle. **longtabu** can be used just like **tabu**.
- Vertical lines can be used whatever the catcode of **|** is.
- **\savetabu** reports saved informations in the .log ([debugshow](#) option).
- **\savetabu ... \usetabu** now restores the **\halign** preamble rather than the **tabu** preamble! **\preamble** can be use in the **tabu** preamble to restore a **tabu** preamble.
- **\tabucline** is more robust with “special” preambles containing **>** or **<** tokens. **\tabucline** now takes care of **\arrayrulecolor** (package **colortbl**).
- **enumitem-zref** package has been added to the documentation (see the link [point 1](#))
- Optimisation of some parts of the code.

**[2010/11/22 v1.4]**

- Compatibility improvement with **linegoal** for the syntax:  
**\begin{tabu}** to **\linegoal{...}**
- Hyper footnotes now work correctly.
- Fix a bug when using colored vertical lines in **tabu** in math mode.
- Fix a bug with vertical lines and **colortbl \arrayrulecolor** specification.
- Fix a compatibility bug with **arydshln**: when nesting a tabular that use vertical dashed lines (**arydshln**) inside **tabu spread** with **X** columns.

**[2010/11/18 v1.3]**

- Fix a bug that may appear in **\tabucline** depending on the preamble due to arbitrary **\countdef**.
- Improvement in the use of **\everycr**: no **\global** stuff. Thus bug fixed when nesting **tabu** inside **\AMS-align** environment for example. Same issue with **\rowfont** which now works without global modification of **\everycr**.
- No phantom line is added to **tabu** but a command **\tabuphantomline** is provided for this purpose (required with **\multicolumn** in some cases).
- Improvement on vertical alignment.
- To do: an example file to test a wide range of possibilities...
- Documentation.

[2010/11/15 v1.2]

- Improvement in parameters parsing for optional parameters (`|` and `\tabucline`).
- Modification / optimization in `\tabu@prepnext@tok`.
- Modification of `\tabucline` to get better results with `m` columns (`X[m]`) and also when `\minrowclearance > 0` (package `colortbl`).

[2010/10/28 v1.1]

- First version.

## 12 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

`\@arrayright` 65, 592, 712, 1603, 1609, 1618, 1622  
`\@arstrutbox` . 102, 172, 173, 175, 178, 179,  
 181, 606, 607, 869, 912, 918, 924, 985,  
 986, 1010, 1250, 1251, 1261, 1262, 1340, 1644  
`\@chclass` . . . . . 1432, 1433  
`\@currentHref` . . . . . 425  
`\@currentlabelname` . . . . . 426  
`\@currenvir` . . . . . 246  
`\@defaultunits` . . 1019, 1020, 1027, 1046, 1048  
`\@elt` . . . . . 598, 599  
`\@finalstrut` . . . . . 869, 1244, 1277, 1644  
`\@firstoftwo` . . . . . 347, 845  
`\@flushglue` . . . . . 224, 227, 228, 855, 1483  
`\@footnotemark` . . . . . 398  
`\@footnotetext` . . . . . 306, 591  
`\@fornoop` . . . . . 1384, 1403, 1419  
`\@gobbletwo` . . . . . 823  
`\@halignto` . . . . . 134, 201, 214, 584, 714, 1301  
`\@ifnextchar` . . . . . 19, 105  
`\@let@token` 326, 328, 329, 330, 331, 332, 482,  
 485, 486, 487, 488, 489, 490, 491, 492,  
 493, 494, 495, 496, 497, 498, 499, 500, 501  
`\@m` . . . . . 797  
`\@makeoother` . . . . . 1142  
`\@mkpream` . . . . . 239  
`\@mpfn` . . . . . 386, 395, 404  
`\@multicnt` . . . . . 981, 987,  
 991, 992, 994, 999, 1000, 1002, 1003, 1007  
`\@normalcr` . . . . . 309, 1595  
`\@preamble` . . . . . 170,  
 613, 619, 623, 1303, 1604, 1610, 1619  
`\@preamerr` . . . . . 1652  
`\@preamerror` . . . . . 1652  
`\@rightskip` . . . . . 1482  
`\@sanitize` . . . . . 1135  
`\@secondoftwo` . . . . . 848  
`\@spaces` . . . . . 792  
`\@sptoken` . . . 219, 330, 498, 1095, 1153, 1187  
`\@tempcnta` . . . . . 586, 1382, 1390  
`\@tempdimb` . . . . . 1051, 1054,  
 1059, 1068, 1069, 1071, 1073, 1080, 1084,  
 1251, 1254, 1262, 1271, 1337, 1339, 1342  
`\@tempdimc` 1064, 1069, 1070, 1071, 1074,  
 1076, 1077, 1086, 1087, 1260, 1262, 1276

`\@temptokena` 360, 374, 380, 899, 1115, 1118,  
 1145, 1161, 1165, 1196, 1199, 1440, 1441  
`\@thefnmark` . . . . . 396, 406  
`\@trivlist` . . . . . 1594, 1595  
`\@undefined` . . . . . 6, 573  
`\@whilenum` . . . . . 920, 987, 1007  
`\@whilesw` . . . . . 720  
`\@xfootnote` . . . . . 307, 391, 591  
`\@xfootnotetext` . . . . . 306  
`\{` . . . . . 1142  
`\}` . . . . . 1142  
`\^` . . . . . 644, 1134, 1135, 1136  
`\_` . . . . . 1136

### A

`\abovetabulinesep` . . . . . 13, 93, 301,  
 606, 1216, 1219, 1220, 1252, 1253, 1263, 1265  
`\adl@act@@endpbox` . . . . . 1651  
`\adl@act@endpbox` . . . . . 1650  
`\adl@arrayrestore` . . . . . 1602, 1608  
`\adl@box` . . . . . 1646  
`\adl@class@start` . . . . . 1628  
`\adl@colhtdp` . . . . . 1646  
`\adl@dashgapcolor` . . . . . 1637, 1639  
`\adl@endarray` . . . . . 1602, 1608  
`\adl@leftrulefalse` . . . . . 1631  
`\adl@ncol` . . . . . 670  
`\adl@vlineL` . . . . . 1637  
`\adl@vlineR` . . . . . 1639  
`\adl@xarraydashrule` . . . . . 1649  
`\afterassignment` . . . . . 20, 21, 256,  
 343, 535, 899, 1045, 1204, 1216, 1571, 1578  
`\aftergroup` . . . . . 157, 205, 206,  
 207, 210, 236, 239, 260, 264, 291, 324,  
 856, 1203, 1215, 1309, 1318, 1552, 1574, 1583  
`\array` . . . . . 267  
`\arraybackslash` . . . . . 854, 1488, 1491,  
 1495, 1498, 1502, 1505, 1509, 1511, 1590  
`\arraycolsep` . . . . . 365  
`\arrayrulewidth` . . . . . 367, 809,  
 945, 960, 1050, 1053, 1057, 1127, 1129, 1634  
`\arraystretch` . . . . . 181, 372, 815, 1341, 1342  
`\AtEndOfPackage` . . . . . 6, 41, 44, 49, 52

<b>B</b>		<b>H</b>	
<code>\bcolumn</code>	862	<code>\H@@footnotetext</code>	418
<code>\begin</code>	4, 4	<code>\hbadness</code>	586
<code>\belowtabulinesep</code>	93, 301, 607, 1219, 1220, 1254, 1255, 1271, 1273	<code>\hfilneg</code>	1547
<b>C</b>		<code>\hfuzz</code>	290, 587
<code>\c@footnote</code>	392	<code>\ht</code>	173, 179, 918, 986, 1248, 1250, 1253, 1261, 1265, 1266, 1268, 1336, 1577, 1586
<code>\c@page</code>	453	<code>\Hy@footnote@currentHref</code>	409
<code>\catcode</code>	11, 13, 644, 1134, 1135, 1136, 1138, 1139	<code>\Hy@raisedlink</code>	422
<code>\cellspacetoplimit</code>	857	<code>\hyper@@anchor</code>	420, 423
<code>\Centering</code>	512, 1495, 1498	<b>I</b>	
<code>\centering</code>	307, 508, 512, 1465, 1467, 1591	<code>\ifadl@inactive</code>	1633
<code>\chardef</code>	102	<code>\ifadl@leftrule</code>	1636
<code>\cl@ckpt</code>	599	<code>\ifadl@zwvrule</code>	1633
<code>\color</code>	819, 1107, 1358, 1366	<code>\ifcase</code>	301, 328, 485, 571, 677, 902, 994, 999, 1049, 1093, 1152, 1169, 1292, 1390, 1414
<code>\color@</code>	1106	<code>\ifcat</code>	164, 241, 501, 959, 1049, 1096, 1231
<code>\color@b@x</code>	311	<code>\ifcsname</code>	432, 720, 839, 948, 1032, 1106, 1359
<code>\color@begingroup</code>	857	<code>\ifdim</code>	171, 177, 211, 292, 296, 301, 351, 448, 543, 546, 564, 565, 606, 607, 677, 678, 680, 681, 690, 694, 698, 702, 713, 721, 724, 727, 738, 741, 745, 749, 764, 846, 871, 875, 877, 878, 1068, 1069, 1070, 1071, 1072, 1073, 1074, 1076, 1080, 1084, 1086, 1252, 1254, 1263, 1271, 1293, 1311, 1330, 1516, 1545
<code>\color@bgroup</code>	1569, 1574	<code>\iffalse</code>	568
<code>\color@block</code>	1577	<code>\ifHy@nesting</code>	419
<code>\color@egroup</code>	1570, 1575	<code>\ifin@</code>	66, 464, 837, 845, 847, 1419
<code>\copy</code>	1031	<code>\ifnum</code>	26, 31, 568, 605, 618, 682, 701, 711, 779, 795, 796, 797, 799, 800, 918, 977, 982, 991, 994, 999, 1002, 1013, 1021, 1022, 1023, 1024, 1238, 1243, 1329, 1351, 1363, 1381, 1399, 1405, 1432, 1433, 1515, 1520, 1541, 1546, 1628, 1629, 1630
<code>\crrc</code>	670, 911, 1617	<code>\iftabu@cellright</code>	1378, 1395, 1410
<code>\CT@drsc@</code>	362, 818	<code>\iftabu@colortbl</code>	255, 360, 373, 816, 913, 926, 944, 946, 971, 1037, 1065, 1124, 1514, 1539
<code>\currentgrouplevel</code>	1227, 1230, 1231, 1235	<code>\iftabu@firstcline</code>	97, 986, 1078, 1126
<code>\currentgrouptype</code>	1238, 1243, 1515, 1520	<code>\iftabu@measuring</code>	97, 440, 637, 663, 707
<b>D</b>		<code>\iftabu@negcoef</code>	97, 205, 571, 676, 831, 1288
<code>\d@llarbegin</code>	1239	<code>\iftabu@spread</code>	97, 446, 571, 634, 660, 844
<code>\d@llarend</code>	164, 241, 1249	<code>\ignorespaces</code>	1152, 1186
<code>\delarray_(package_option)</code>	48	<code>\immediate</code>	258, 260, 264
<code>\detokenize</code>	821, 948, 951, 1032, 1033, 1116	<code>\in@</code>	65, 464, 836, 1418
<code>\dimen@</code>	675, 680, 698, 716, 1586, 1587	<code>\in@false</code>	846
<code>\doublerulesep</code>	368, 810	<b>J</b>	
<code>\dp</code>	175, 181, 1248, 1251, 1255, 1262, 1337, 1577, 1587	<code>\justifying</code>	514, 1509, 1511
<b>E</b>		<b>L</b>	
<code>\ecolumn</code>	862	<code>\lastbox</code>	295
<code>\endarray</code>	65, 66, 267, 281, 659, 1605, 1611, 1623	<code>\lastskip</code>	1540, 1545, 1546
<code>\endlinechar</code>	222, 225, 931, 1137	<code>\leaders</code>	1125
<code>\endlongtable</code>	274	<code>\leftskip</code>	1482
<code>\endlongtabu</code>	274	<code>\light_(package_option)</code>	51
<code>\endlrbox</code>	1249	<code>\linebreak</code>	1136
<code>\endtabu</code>	266	<code>\linegoal</code>	46, 452
<code>\endtabular</code>	268	<code>\linegoal_(package_option)</code>	42
<code>\everycr</code>	314, 898, 899	<code>\linewidth</code>	42, 46, 286, 292, 296, 451, 589
<code>\everyeof</code>	1143	<code>\llap</code>	1264, 1272
<code>\everypar</code>	293, 294, 1244	<code>\LNGL@setlinegoal</code>	346, 347
<code>\extracolsep</code>	212, 224, 227, 228	<code>\longtable</code>	274
<code>\extrarowdepth</code>	93, 177, 181, 370, 812, 1207, 1208, 1339	<code>\longtabu</code>	71, 272, 277
<code>\extrarowheight</code>	13, 369, 811, 918, 1038, 1082, 1083, 1129, 1130, 1204, 1207, 1208, 1338	<code>\loop</code>	989
<code>\extrarowsep</code>	13, 1202	<b>F</b>	
<code>\extratabsurround</code>	371, 813	<code>\footnotetext</code>	386, 390
<b>F</b>		<code>\futurelet</code>	217, 326, 344, 482, 953, 1091, 1098, 1099, 1149
<code>\GenericError</code>	262, 590	<b>G</b>	
<code>\gluestretchorder</code>	1541, 1546	<code>\GenericError</code>	262, 590

`\lower` ..... 1038  
`\lrbox` ..... 1239  
`\LT@bchunk` ..... 616, 620, 625  
`\LT@echunk` ..... 630  
`\LT@get@widths` ..... 632  
`\LT@output` ..... 626  
`\LTchunksize` ..... 624

## M

`\maxdimen` ..... 587  
`\message` ..... 27, 31  
`\minrowclearance` .... 373, 814, 914, 915, 927  
`\multicolumn` ..... 56, 288, 289  
`\multispan` ..... 239, 251, 252

## N

`\narrowragged` ..... 854  
`\NC@find` ..... 135, 166, 193,  
     197, 200, 224, 227, 228, 243, 376, 821, 939  
`\NC@rewrite@X` ..... 155  
`\newcount` ..... 81, 82, 83, 84, 85, 86  
`\newdimen` 87, 88, 89, 90, 91, 92, 93, 94, 95, 96  
`\newsavebox` ..... 101  
`\newskip` ..... 1534  
`\newtoks` ..... 383  
`\noalign` ..... 670, 901, 911,  
     923, 977, 985, 1010, 1280, 1351, 1361, 1368  
`\nointerlineskip` ..... 1268, 1276  
`\null` ..... 1136  
`\numexpr` 782, 783, 784, 1202, 1214, 1289, 1390

## O

`\omit` ..... 990  
`\on@line` ..... 453, 778, 825

## P

`\PackageError` ..... 59, 61, 273, 278, 1348  
`\PackageInfo` ..... 1345  
`\PackageWarning` .....  
     ..... 75, 265, 523, 530, 1039, 1325, 1621  
`\parfillskip` ..... 855, 1136, 1483  
`\pdfadjustspacing` ..... 863, 864  
`\pdfelapsedtime` ..... 28, 29, 782, 783, 784  
`\preamble` ..... 15, 144, 150, 160  
`\preamble␣(private␣column␣type)` ..... 144  
`\prepnext@tok` ..... 53, 313, 1379, 1419  
`\prevdepth` ..... 1260  
`\ProcessOptions` ..... 63  
`\protected` ..... 259, 263  
`\protected@xdef` ..... 396, 405

## Q

`\q` ..... 950

## R

`\RaggedLeft` ..... 513, 1502, 1505  
`\raggedleft` ... 308, 509, 513, 1471, 1473, 1592  
`\RaggedRight` ..... 511, 1488, 1491  
`\raggedright` . 308, 507, 511, 1459, 1461, 1593  
`\remove@to@nnil` ..... 503, 1101, 1108  
`\repeat` ..... 1006  
`\rightskip` ..... 1482  
`\romannumeral` ..... 950  
`\rowfont` ..... 12, 60, 312, 313, 1351

## S

`\save@decl` ..... 53, 165, 242  
`\savetabu` 14, 355, 589, 710, 825, 826, 1280, 1346  
`\scantokens` ..... 222, 225, 931, 1143  
`\set@color` ..... 1107, 1582  
`\setbox` ..... 172, 178, 295, 606, 607,  
     610, 622, 631, 851, 1242, 1340, 1572, 1579  
`\setlength` ..... 449  
`\span` ..... 1005  
`\stretch` ..... 1515, 1517  
`\strip@pt` ..... 442, 456, 550  
`\strutbox` ..... 1336, 1337

## T

`\tabcolsep` ..... 366  
`\tabu` ..... 246, 266  
`\tabu@` ..... 267, 268, 274, 357  
`\tabu@@begin` ..... 352, 354, 358  
`\tabu@@cellalign` ..... 1356, 1373  
`\tabu@@celleft` ..... 1354, 1370  
`\tabu@@cellralign` ..... 1357, 1374  
`\tabu@@cellright` ..... 1355, 1371  
`\tabu@@rewrite@X` ..... 463, 852  
`\tabu@@tabudecimal` ..... 1167  
`\tabu@add` ..... 992, 1000, 1015  
`\tabu@adl@act@endpbox` 1643, 1647, 1650, 1651  
`\tabu@adl@endTRIAL` ..... 70, 658, 670  
`\tabu@adl@fix` ..... 70, 281, 1648, 1653  
`\tabu@adl@xarraydashrule` .. 1627, 1642, 1649  
`\tabu@align` 356, 357, 986, 1306, 1307, 1315, 1316  
`\tabu@aligndefault` ..... 286, 290, 352, 1324  
`\tabu@aligndefault@text` ... 1306, 1315, 1324  
`\tabu@arith` ..... 635, 661, 671  
`\tabu@arith@negcoef` ..... 676, 718, 733  
`\tabu@arrayleft@measure` ..... 269, 275, 581  
`\tabu@arrayleftmeasure` ..... 269, 603  
`\tabu@arstrutbox` ..... 102, 924, 1010  
`\tabu@begin` ..... 326  
`\tabu@begin@color@b@x` ... 1578, 1581, 1584  
`\tabu@begin@fbox` ..... 1571, 1574  
`\tabu@box` 101, 610, 622, 633, 677, 678, 687,  
     692, 699, 701, 738, 740, 749, 760, 765,  
     769, 774, 851, 869, 870, 873, 878, 879,  
     886, 1239, 1242, 1248, 1250, 1251, 1253,  
     1255, 1256, 1261, 1265, 1266, 1268, 1270  
`\tabu@C@linesep` ..... 1214, 1220, 1223, 1224  
`\tabu@C@rowsep` ..... 1202, 1208, 1211, 1212  
`\tabu@cell@align` ..... 1449, 1458,  
     1464, 1470, 1476, 1487, 1494, 1501, 1508  
`\tabu@cell@C` ..... 1493, 1499, 1530  
`\tabu@cell@c` ..... 1457  
`\tabu@cell@J` ..... 1507, 1512, 1531  
`\tabu@cell@j` ..... 1457  
`\tabu@cell@L` ..... 1486, 1492, 1528  
`\tabu@cell@l` ..... 1457  
`\tabu@cell@R` ..... 1500, 1506, 1529  
`\tabu@cell@r` ..... 1457  
`\tabu@cellfont` ..... 1358, 1361, 1372, 1444  
`\tabu@celllalign` ..... 302,  
     303, 1356, 1373, 1425, 1444, 1450, 1451  
`\tabu@celleft` ..... 1185, 1354, 1361,  
     1362, 1370, 1426, 1444, 1454, 1455, 1459,  
     1465, 1471, 1477, 1489, 1496, 1503, 1509  
`\tabu@cellralign` ..... 302,  
     304, 1357, 1374, 1422, 1444, 1452, 1453

<code>\tabu@cellright</code>	1355, 1371, 1422, <a href="#">1444</a>	<code>\tabu@global@X</code>	202, 235, 236, 282, 441, 455, 583, 599, 609, 620, 624, 708
<code>\tabu@cellrightfalse</code>	1388, 1396, 1411	<code>\tabu@global@X@nested</code>	232, 234
<code>\tabu@cellrighttrue</code>	1402	<code>\tabu@global@Xnested</code>	204, 232, 237, 605, 618
<code>\tabu@cellskip</code>	1516, 1534, 1540, 1541, 1542	<code>\tabu@gobble@pt</code>	105, 106
<code>\tabu@centering</code>	307, 1591	<code>\tabu@gobbleopt</code>	<a href="#">104</a> , 467
<code>\tabu@checklast</code>	186, 188, 196, 198	<code>\tabu@gobblespace</code>	<a href="#">104</a> , 220, 335, 1099, 1157
<code>\tabu@ccline</code>	978, 979, 980, 1014	<code>\tabu@Grestore</code>	1211, 1223, <a href="#">1226</a>
<code>\tabu@cnt</code>	<a href="#">81</a> , 535, 538, 580, 672, 682, 701, 799	<code>\tabu@Growsep</code>	1203, 1210
<code>\tabu@collect</code>	614, 627, <a href="#">644</a>	<code>\tabu@Gsave</code>	1208, 1220, <a href="#">1226</a>
<code>\tabu@collectbody</code>	644, 645, 652, 654	<code>\tabu@halignto</code>	203, 214, 341, 342, 351, 376, 460, 754, 1301
<code>\tabu@color</code>	958, 970, 972, 1026, 1045, 1058, 1065, 1066, 1075, 1107, 1115, 1118	<code>\tabu@header</code>	686, 758, 772, 799
<code>\tabu@color@b@x</code>	311, 1576, 1580	<code>\tabu@hfil</code>	1535, 1555, 1561
<code>\tabu@colorleavevmode</code>	1358, 1365	<code>\tabu@hfill</code>	1536, 1562
<code>\tabu@colorORI</code>	1358, 1365, 1366	<code>\tabu@hfuzz</code>	290, 677, 688, 776, 778
<code>\tabu@colortblfalse</code>	68	<code>\tabu@hskip</code>	1537, 1563
<code>\tabu@colortbltrue</code>	68	<code>\tabu@Hy@footnotetext</code>	407, 417, 430
<code>\tabu@colortest</code>	1045, <a href="#">1091</a>	<code>\tabu@Hy@ftntext</code>	<a href="#">392</a>
<code>\tabu@dashgap</code>	976, 1052, 1055, 1069	<code>\tabu@Hy@ftntxt</code>	392, 398, 400, 416
<code>\tabu@debug</code>	31, 32, 36, 880, 1011	<code>\tabu@Hy@xfootnote</code>	<a href="#">392</a>
<code>\tabu@decimal</code>	1145, 1199	<code>\tabu@ignorehfil</code>	1460, 1466, 1472, 1490, 1497, 1504, <a href="#">1551</a>
<code>\tabu@defaultleaders</code>	1028, 1042, 1123, 1124	<code>\tabu@immediate</code>	258, 261, 265
<code>\tabu@defleaders</code>	1029, 1030, 1031, 1036	<code>\tabu@islast</code>	158, 190, 192
<code>\tabu@dimexpr</code>	153, 473, 560, 563, 566, 836, 845, 848	<code>\tabu@justify</code>	510, 514, 1477, 1479, 1481, 1484
<code>\tabu@do@color@b@x</code>	1577, 1588	<code>\tabu@lastnoop</code>	1403, 1418
<code>\tabu@elapsedtime</code>	30, 779	<code>\tabu@lastrewrite</code>	191, 199, 216
<code>\tabu@end@color@b@x</code>	1583, 1585, 1589	<code>\tabu@lasttry</code>	580, 678, 699, 701
<code>\tabu@end@fbox</code>	1574, 1575	<code>\tabu@leaderfill</code>	1038, 1088, 1132, 1133
<code>\tabu@endarray</code>	66, 281, 1601, 1607, 1616	<code>\tabu@linecolor</code>	1091, 1092, 1098, 1099, 1103
<code>\tabu@endenvir</code>	641, 648, 667	<code>\tabu@linedash</code>	975, 1044, 1071
<code>\tabu@endrewrite</code>	185, 197, 198	<code>\tabu@linegoal@target</code>	344, 345, 349
<code>\tabu@endTRIAL</code>	611, <a href="#">657</a>	<code>\tabu@linegoaltarget</code>	343, 344
<code>\tabu@everycr</code>	314, 898, 900	<code>\tabu@lineoff</code>	1052, 1055, 1060, 1063, 1090
<code>\tabu@extracol@sep</code>	217, 218, 220, 231	<code>\tabu@lineon</code>	1044, 1047, 1062
<code>\tabu@extracolsep</code>	212, 217	<code>\tabu@lines</code>	55, 161, 240, <a href="#">931</a>
<code>\tabu@fbox</code>	311, 1567, 1573	<code>\tabu@lines@</code>	932, 934, 937
<code>\tabu@first@ccline</code>	897, 902, 903, 904, 906, 928	<code>\tabu@linestyle</code>	1111, 1112, 1121
<code>\tabu@firstccline</code>	55	<code>\tabu@long@arrayleft@measure</code>	275, <a href="#">616</a>
<code>\tabu@firstcclinefalse</code>	905	<code>\tabu@long@endTRIAL</code>	625, 629, 643
<code>\tabu@firstcclinetrue</code>	896	<code>\tabu@make@arstrutbox</code>	1304, 1335, 1344
<code>\tabu@fix@arrayright</code>	67, 1596, 1626	<code>\tabu@makeatletter</code>	1135
<code>\tabu@flush</code>	1460, 1465, 1466, 1471, 1490, 1496, 1497, 1503, 1513, 1524	<code>\tabu@maybeicolor</code>	966, <a href="#">1044</a>
<code>\tabu@footnotetext</code>	<a href="#">384</a>	<code>\tabu@measure@endpbox</code>	<a href="#">834</a>
<code>\tabu@footnotes</code>	291, <a href="#">383</a> , 385, 387, 389, 402, 412	<code>\tabu@measure@startpbox</code>	578, 673, 763, <a href="#">834</a>
<code>\tabu@footnotetext</code>	306, 384, 434	<code>\tabu@measuringfalse</code>	315, 637, 663
<code>\tabu@fornoopORI</code>	1384, 1419	<code>\tabu@measuringtrue</code>	445, 683, 690, 697, 756
<code>\tabu@G@linesep</code>	1220, 1222, 1223, 1225	<code>\tabu@message</code>	27, 34, 450, 454, 672, 680, 681, 682, 686, 726, 737, 744, 745, 748, 749, 824
<code>\tabu@G@rowsep</code>	1208, 1210, 1211, 1213	<code>\tabu@message@arith</code>	672, <a href="#">758</a>
<code>\tabu@GenericError</code>	259, 590	<code>\tabu@message@etime</code>	30, 780
<code>\tabu@GenericErrorORI</code>	261, 262	<code>\tabu@message@negcoef</code>	726, <a href="#">758</a>
<code>\tabu@get@decimal</code>	1161, 1189	<code>\tabu@message@reached</code>	680, 681, 682, 772
<code>\tabu@get@decimalspace</code>	1164, 1191	<code>\tabu@message@save</code>	27, 35, 1321
<code>\tabu@getdecimal</code>	1155, <a href="#">1168</a>	<code>\tabu@message@spread</code>	737, <a href="#">758</a>
<code>\tabu@getdecimal@</code>	1146, 1149, 1155	<code>\tabu@minrowclearance</code>	914, 927
<code>\tabu@getdecimal@ignorespaces</code>	1146, 1151, 1160	<code>\tabu@msg@align</code>	748, 759, 760, 761, 762, 765, 768, 769, 770, 771, 773, 774, 775, 794
<code>\tabu@getline</code>	1034, <a href="#">1044</a>	<code>\tabu@multic@lumn</code>	247, 251, 254
<code>\tabu@getlinecolor</code>	961, 1100, 1104, 1109	<code>\tabu@multicolumn</code>	56, 245, 250, 289
<code>\tabu@getlinespec</code>	981, <a href="#">1026</a> , 1113	<code>\tabu@multicolumnORI</code>	248, 253, 288
<code>\tabu@gettarget</code>	341, 342, 343	<code>\tabu@naturalX</code>	<a href="#">87</a> , 321, 584, 735, 740, 771, 841, 870, 871, 872
<code>\tabu@Glinesep</code>	1215, 1222	<code>\tabu@naturalX@max</code>	323, 575, 735, 871, 872, 894
<code>\tabu@global@temp</code>	156, 157, 1385, 1386, 1393, 1394, 1440, 1441		



<code>\tabu@naturalX@min</code>	322, 576, 736, 875, 876, 893
<code>\tabu@nbcols</code>	81, 670, 920, 1007, 1017, 1021, 1023, 1024, 1329, 1383, 1386, 1391, 1392, 1394, 1399, 1405, 1407, 1408, 1429
<code>\tabu@NC@list</code>	163, 169, 185, 316
<code>\tabu@NC@rewrite@X</code>	155
<code>\tabu@negcoefficient</code>	315
<code>\tabu@negcoefficienttrue</code>	205, 543
<code>\tabu@nested</code>	81, 233, 285, 319, 324, 711, 779, 800, 888, 890, 892, 918
<code>\tabu@new@columntype</code>	111, 116, 119, 130
<code>\tabu@newcolumntype</code>	110, 151, 184, 197, 198, 238, 935
<code>\tabu@next</code>	200, 212, 215, 224, 227, 228, 229, 1098, 1099, 1100, 1102
<code>\tabu@noadjustspacing</code>	855, 864, 866
<code>\tabu@nocellspace</code>	853, 862
<code>\tabu@nocline</code>	54, 61
<code>\tabu@nohfil</code>	1552, 1554, 1556, 1560
<code>\tabu@nolongtabu</code>	71, 278
<code>\tabu@norowfont</code>	54, 59
<code>\tabu@nowrite</code>	256, 590
<code>\tabu@preamble</code>	1314, 1318
<code>\tabu@prep@TRIAL</code>	206, 567
<code>\tabu@prepare@save</code>	361, 362, 378, 382
<code>\tabu@preparesave</code>	355, 359
<code>\tabu@prepnext@tok</code>	53, 313, 1378
<code>\tabu@prepnext@tokORI</code>	1379, 1416
<code>\tabu@preptokenleft</code>	1400, 1425, 1427
<code>\tabu@preptokenright</code>	1397, 1412, 1420, 1424
<code>\tabu@printdecimal</code>	1192, 1195, 1201
<code>\tabu@private@columntype</code>	121, 128, 131
<code>\tabu@privatecolumns</code>	129, 132, 159
<code>\tabu@privatecolumntype</code>	110, 133, 136, 144
<code>\tabu@pushbegins</code>	646, 655, 656
<code>\tabu@raggedleft</code>	308, 1592
<code>\tabu@raggedright</code>	308, 1593
<code>\tabu@removehfil</code>	1459, 1465, 1471, 1488, 1495, 1502, 1534
<code>\tabu@restoreeverycr</code>	898, 899
<code>\tabu@rewrite@multicolumn</code>	238
<code>\tabu@rewrite@X</code>	133, 140, 438, 588, 852
<code>\tabu@rewrite@Xnested</code>	464, 588
<code>\tabu@rewrite@Xrestore</code>	140, 464
<code>\tabu@rewritefirst</code>	151, 167, 316, 357
<code>\tabu@rewritefirst_(new_columntype)</code>	151
<code>\tabu@rewritelast</code>	185, 193, 197
<code>\tabu@rewritelast_(new_columntype)</code>	184
<code>\tabu@rewritemiddle</code>	163, 166, 184, 187
<code>\tabu@Rlinesep</code>	1222, 1223
<code>\tabu@row@font</code>	1351, 1352
<code>\tabu@rowfont</code>	54, 312, 1351
<code>\tabu@rowfont@reset</code>	1353, 1367
<code>\tabu@rowfontreset</code>	908, 1353, 1369, 1377
<code>\tabu@Rrowsep</code>	1210, 1211
<code>\tabu@sanitizetext</code>	283, 310, 1134
<code>\tabu@save@decl</code>	53, 107, 165, 242
<code>\tabu@savecounters</code>	585, 597, 600
<code>\tabu@savedparam</code>	364, 829, 1285
<code>\tabu@savedpream</code>	376, 830, 1286
<code>\tabu@savedpreamble</code>	170, 613, 619, 623, 1284
<code>\tabu@saveerr</code>	137, 139, 145, 147, 1345
<code>\tabu@savepreamble</code>	154, 156, 168, 183
<code>\tabu@savetok</code>	1429, 1431
<code>\tabu@savetok</code>	1397, 1399, 1402, 1406, 1408, 1428, 1430, 1443
<code>\tabu@savewarn</code>	1281, 1282, 1345
<code>\tabu@savewd</code>	831, 1289, 1290, 1328, 1332, 1334
<code>\tabu@scandecimal</code>	1147, 1149, 1150, 1157, 1162, 1165
<code>\tabu@setlinesep</code>	1216, 1218, 1221
<code>\tabu@setrowsep</code>	1204, 1206, 1209
<code>\tabu@settarget</code>	270, 276, 326
<code>\tabu@setup</code>	270, 276, 280
<code>\tabu@setup@everycr</code>	56, 207, 210, 612, 619, 623, 895
<code>\tabu@setup@saveglobal</code>	284, 318, 325
<code>\tabu@setup@TRIAL</code>	569, 583
<code>\tabu@skipdecimal</code>	1150, 1156, 1158, 1190
<code>\tabu@spaces</code>	763, 777, 780, 786, 792, 880
<code>\tabu@spread</code>	337, 342
<code>\tabu@spreadarith</code>	634, 660, 734
<code>\tabu@spreadfalse</code>	315, 756
<code>\tabu@spreadheader</code>	767, 803
<code>\tabu@spreadtarget</code>	87, 446, 447, 459, 738, 739, 741, 742, 745, 768
<code>\tabu@spreadtrue</code>	342
<code>\tabu@start</code>	81, 982, 987, 1019, 1021, 1022
<code>\tabu@start@stop</code>	1017, 1018, 1025
<code>\tabu@startpboxORI</code>	577, 674, 753, 835
<code>\tabu@startstop</code>	981, 1017
<code>\tabu@starttime</code>	29, 782, 783, 784
<code>\tabu@stop</code>	81, 982, 994, 999, 1002, 1020, 1023, 1024
<code>\tabu@tabudecimal</code>	310, 1144, 1148, 1167
<code>\tabu@target</code>	87, 211, 327, 343, 351, 376, 446, 448, 449, 453, 458, 570, 677, 688, 692, 710, 714, 738, 741, 742, 744, 745, 748, 749, 754, 755, 761, 765, 775, 828, 1293, 1296, 1311, 1312
<code>\tabu@temp</code>	189, 190, 192, 217, 219, 223, 226, 344, 346, 474, 478, 534, 538, 550, 558, 560, 722, 724, 725, 849, 854, 882, 939, 943, 949, 950, 953, 959, 969, 1091, 1093, 1094, 1095, 1096, 1098, 1099, 1149, 1152, 1153, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1196, 1197, 1198, 1590, 1591, 1592, 1593
<code>\tabu@timerstart</code>	29, 37, 287
<code>\tabu@titles</code>	454, 800
<code>\tabu@to</code>	336, 341
<code>\tabu@tr@cing</code>	23, 25, 39
<code>\tabu@tracing</code>	20, 21, 22, 24
<code>\tabu@tracing@save</code>	27, 807
<code>\tabu@TRIAL</code>	603, 621, 638, 649, 664
<code>\tabu@TRIAL@hook</code>	595, 601, 602
<code>\tabu@trivlist</code>	309, 1594
<code>\tabu@usetabu</code>	1299, 1309
<code>\tabu@vcenter</code>	103, 925
<code>\tabu@verb</code>	1135, 1141
<code>\tabu@verticalmeasure</code>	303, 593, 1237
<code>\tabu@verticalsp@lcr</code>	1240, 1248
<code>\tabu@verticalsp@pmb</code>	1245, 1258
<code>\tabu@verticalspacing</code>	304, 594, 1240, 1245
<code>\tabu@vline</code>	940, 945, 973
<code>\tabu@warn</code>	263, 687, 695
<code>\tabu@warn@cellspace</code>	72, 74, 80
<code>\tabu@warn@usetabu</code>	1294, 1325

<code>\tabu@wd</code> . . .	564, 565, 721, 724, 727, 731, 788, 846, 877, 878, 883, 884, 888, 1330, 1331
<code>\tabu@wddef</code> . . . . .	822, 879, 889, 1331
<code>\tabu@wdlet</code> . . . . .	470, 545, 891
<code>\tabu@write</code> . . . . .	257, 261, 265
<code>\tabu@X@cols</code> <a href="#">81</a> , 152, 202, 320, 439, 442, 469, 470, 473, 545, 560, 572, 573, 574, 719, 720, 721, 723, 724, 726, 727, 729, 838, 839, 840, 874, 877, 878, 879, 881, 883, 884	
<code>\tabu@X@sum</code> . . . . .	<a href="#">90</a> , 442, 443, 456, 457, 472, 476, 538, 543, 544, 546, 550, 675, 680, 681, 684, 690, 692, 693, 698, 713, 716, 725, 762, 764, 766, 873
<code>\tabu@Xalign</code> . . . . .	507, 508, 509, 510, 511, 512, 513, 514, 522, 528
<code>\tabu@Xarg</code> . . . . .	438, <a href="#">468</a>
<code>\tabu@Xc@ef</code> . . . . .	535, 537, 540
<code>\tabu@Xcoef</code> . . . . .	516, 518, 534, 536
<code>\tabu@Xfinish</code> . . . . .	604, 617, 640, 666
<code>\tabu@xfootnote</code> . . . . .	307, 391, 435
<code>\tabu@xfootnotetext</code> . . . . .	306, 388
<code>\tabu@xleaders</code> . . . . .	983, 997, 1026, 1028, 1037, 1042, 1075, 1114, 1119, 1123
<code>\tabu@Xmath</code> . . . . .	478, 517, 552, 556
<code>\tabu@Xparse</code> . . . . .	<a href="#">468</a>
<code>\tabu@Xparsespace</code> . . . . .	483, 515
<code>\tabu@Xtest</code> . . . . .	482, 484, 521
<code>\tabu@Xtype</code> . . . . .	504, 505, 506, 529, 533
<code>\tabucline</code> . . . . .	8, 54, 62, 313, <a href="#">975</a>
<code>\tabucline*</code> . . . . .	<a href="#">9</a>
<code>\tabucline@warn</code> . . . . .	1039, 1043, 1058
<code>\tabucolX</code> <a href="#">90</a> , 465, 466, 473, 474, 550, 570, 684, 690, 692, 694, 696, 702, 709, 724, 736, 740, 755, 759, 770, 773, 790, 827, 1297, 1302	
<code>\tabucolX@align</code> . . . . .	477, 523, 525, 526, 549, 553
<code>\tabucolX@do</code> . . . . .	<a href="#">468</a>
<code>\tabucolX@error</code> . . . . .	<a href="#">90</a> , 696
<code>\tabucolX@rw</code> . . . . .	135, 465, 466, 467, 473, 477, 530, 532, 542, 551, 555, 558, 559, 560
<code>\tabudecimal</code> . . . . .	<a href="#">11</a> , 310, <a href="#">1144</a>
<code>\tabudefaultrighttarget</code> . . . . .	42, 46, 286, 292, 449, 451, 452, 589
<code>\tabuDisableCommands</code> . . . . .	<a href="#">601</a>
<code>\tabular</code> . . . . .	268
<code>\tabulinesep</code> . . . . .	<a href="#">14</a> , 78, <a href="#">1214</a>
<code>\tabulinestyle</code> . . . . .	<a href="#">9</a> , <a href="#">1110</a>
<code>\tabuphantomline</code> . . . . .	<a href="#">7</a> , <a href="#">911</a>
<code>\tabusaveX</code> . . . . .	826
<code>\tabustrutrule</code> <a href="#">14</a> , <a href="#">93</a> , 171, 174, 180, 1267, 1275	
<code>\tabuvline@arg</code> . . . . .	938, <a href="#">941</a>
<code>\tabuvline@argi</code> . . . . .	953, 957, 964
<code>\tabuvline@argii</code> . . . . .	962, 965, 967
<code>\tabuvline@argiii</code> . . . . .	952, 953, 968, 974
<code>\tabuvline@rewrite</code> . . . . .	<a href="#">931</a>
<code>\thempfn</code> . . . . .	396, 406
<code>\toks</code> . . .	108, 1227, 1230, 1231, 1235, 1284, 1285, 1286, 1287, 1290, 1298, 1300, 1303, 1305, 1320, 1422, 1425, 1426, 1432, 1433
<code>\tracingtabu</code> . . . . .	<a href="#">18</a>
<code>\tracingtabu_ (debugshow_ option)</code> . . . . .	<a href="#">19</a>
<code>\ttfamily</code> . . . . .	1140
<b>U</b>	
<code>\unhbox</code> . . . . .	631
<code>\unkern</code> . . . . .	1540
<code>\unless</code> . . . . .	1435
<code>\unskip</code> . . . . .	1139, 1545, 1644
<code>\unvbox</code> . . . . .	1270
<code>\usetabu</code> . . .	<a href="#">14</a> , 136, 143, 160, 474, 1326, 1349
<code>\usetabu_ (private_ column_ type)</code> . . . . .	<a href="#">136</a>
<b>V</b>	
<code>\value</code> . . . . .	395, 404, 598
<code>\vbox</code> . . . . .	917, 1243
<code>\vcenter</code> . . . . .	103, 917, 925
<code>\verbatim</code> . . . . .	<a href="#">10</a> , 283, 310
<code>\vskip</code> . . . . .	986, 1268, 1276
<code>\vspace</code> . . . . .	918
<code>\vtop</code> . . . . .	886, 1243
<b>W</b>	
<code>\write</code> . . . . .	257, 260, 264, 590
<b>X</b>	
<code>\X_ (private_ column_ type)</code> . . . . .	<a href="#">133</a>
<code>\xleaders</code> . . . . .	1038, 1075
<b>Z</b>	
<code>\zap@space</code> . . . . .	1105