

# tabu

tabu and longtabu

Flexible  $\text{\LaTeX}$  tabulars

FC

2011/01/19 – version 2.1

1	2	3	4
5	6	7	
8	9		
10			

$\eta$	$\delta$	$\beta$	$\alpha$
$\theta$	$\epsilon$	$\gamma$	
	$\iota$	$\varsigma$	
		$\kappa$	

## 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  $\epsilon$ - $\text{\TeX}$  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 Automatic horizontal lines: <code>\everyrow</code> .....	12
3 Modifying the font and the alignment in one row: <code>\rowfont</code> .....	12
4 Controlling vertical spacing of lines: <code>\extrarowsep</code> and <code>\tabulinesep</code> .....	13
4.1 <code>\extrarowheight</code> <code>\extrarowdepth</code> and <code>\extrarowsep</code> .....	13
4.2 <code>\abovetabulinesep</code> <code>\belowtabulinesep</code> and <code>\tabulinesep</code> .....	14
5 Saving and restoring a <code>tabu</code> : <code>\savetabu</code> , <code>\usetabu</code> and <code>\preamble</code> .....	15
6 Differences between <code>tabu</code> , <code>tabular</code> , <code>tabularx</code> and <code>longtable</code> .....	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`

6.1	Paragraph indentation	16
6.2	Custom environments	16
6.3	Inversion of tokens	16
6.4	Improved process for rewriting columns ( <i>for keen readers</i> )	17
<b>7</b>	<b>The package options</b>	<b>19</b>
7.1	The <code>debugshow</code> package option	19
7.2	The <code>linegoal</code> package option	19
7.3	The <code>delarray</code> package option	19
7.4	The <code>light</code> package option	19
<b>8</b>	<b>Corrections of some bugs (<i>available only inside tabu</i>)</b>	<b>20</b>
8.1	Correction for <code>colortbl</code> and <code>arydshln</code> : compatibility with <code>delarray</code>	20
8.2	Correction for <code>arydshln</code> : <b>@</b> columns	20
<b>9</b>	<b>To do for even better tabus</b>	<b>20</b>
<b>10</b>	<b>IMPLEMENTATION</b>	<b>21</b>
10.1	Identification, requirements and options	21
10.2	Some constants	22
10.3	<code>\tabu</code> , <code>\endtabu</code> , <code>\longtabu</code> and <code>\endlontabu</code>	27
10.4	Footnotes inside <code>tabu</code>	30
10.5	Flow chart of expansion	32
10.6	<code>tabu</code> X column definition	34
10.7	Before trials to reach the target: setup after <code>\@mkpream</code>	36
10.8	One trial after another	37
10.9	Algorithms: Measuring the <code>tabu</code>	39
10.10	<code>\everycr</code> setup and <code>\tabupantomline</code>	43
10.11	Lines inside <code>tabu</code>	44
10.12	Verbatim inside <code>tabu</code> with X columns	48
10.13	Numbers in <code>tabu</code>	48
10.14	<code>\extrarowsep</code> and <code>\tabulinesep</code>	49
10.15	<code>\savetabu</code>	51
10.16	<code>\rowfont</code>	52
10.17	Utilities	58
10.18	Corrections	59
<b>11</b>	<b>References</b>	<b>61</b>
<b>12</b>	<b>History</b>	<b>61</b>
	[2011/01/19 v2.1]	61
	[2011/01/18 v2.0]	61
	[2011/01/15 v1.9]	62
	[2010/12/28 v1.8]	62
	[2010/12/18 v1.7]	62
	[2010/12/07 v1.5]	62
	[2010/11/22 v1.4]	62
	[2010/11/18 v1.3]	63
	[2010/11/15 v1.2]	63
	[2010/10/28 v1.1]	63
<b>13</b>	<b>Index</b>	<b>63</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.&gt;{\$}X&lt;{\$}</i> )
<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 target has been specified (<i>ie.</i> the default target is used).</li> </ul> In this case, <b>tabu</b> behaves as if it was said: <b>{tabu} spread Opt</b> .
<b>\everyrow{code}</b>	Allows to add horizontal lines automatically for every row. The settings can be changed inside the <b>tabu</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 <code>\tabulinestyle</code>
<b>\tabulinestyle{name=spec,...}</b>	Defines a line style for use with <code>\tabucline[name]</code> or with <code> [name]</code>
<b>\tracingtabu</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> .
<b>\tracingtabu=2</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 `ltxtable`.

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  $\frac{5}{7}$  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:

<code>\begin{tabu} to\linewidth{   X[-2.5c]   X[c]   }:</code>	
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 `tabu spread` behaves as if you didn’t give it a target.  
  
The result of this example is the same as if one had written `\begin{tabu}to\linewidth.`

Sherlock Holmes

The “official” web site: <http://www.sherlockholmes.com/>

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 discards 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-
\multicolumn{2}{|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-
\multicolumn{2}{|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:

	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>
	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:

<table><tr><td>Hello</td><td>world</td></tr><tr><td colspan="2"><hr/></td></tr><tr><td>Hello</td><td>world</td></tr></table> <code>\tabucline</code>	Hello	world	<hr/>		Hello	world	<table><tr><td>Hello</td><td>world</td></tr><tr><td colspan="2"><hr/></td></tr><tr><td>Hello</td><td>world</td></tr></table> <code>\tabucline*</code>	Hello	world	<hr/>		Hello	world	<table><tr><td>Hello</td><td>world</td></tr><tr><td colspan="2"><hr/></td></tr><tr><td>Hello</td><td>world</td></tr></table> <code>\firsthline and \hline and \lasthline</code>	Hello	world	<hr/>		Hello	world
Hello	world																			
<hr/>																				
Hello	world																			
Hello	world																			
<hr/>																				
Hello	world																			
Hello	world																			
<hr/>																				
Hello	world																			

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[1$]}`.

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 0pt{ |[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

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

## 2 Automatic horizontal lines: `\everyrow`

`\everyrow`

`\everyrow` can be used to insert horizontal lines automatically:

```
\begin{tabu}.5\linewidth{cX[2c]X} \tabucline[1pt]- \everyrow{\tabucline[on 2pt]-}
This is      &a small example &&of a \textt{tabu}      &&\\
which      &&\automatically &&\inserts      &&\\
a horizontal &&\line after      &&\zeach of its row &&\everyrow{} \\ \tabucline[1pt]-
\end{tabu}
```

This is	a small example	of a tabu
which	automatically	inserts
a horizontal	line after	each of
		its row

`\everyrow` can be used in `longtabu` as well.

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

`\rowfont[⟨alignment⟩]{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 <code>ragged2e</code> 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]|}
\rowfont[c]\bfseries
This & Is
\xpackage{tabu} & \package
\rowfont[r]\itshape
for & \textt{tabu} and \textt{longtabu}
\end{tabu}

```

This	Is
tabu	package
for	tabu and longtabu

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

### 4.1 `\extrarowheight` `\extrarowdepth` and `\extrarowsep`

```

\extrarowheight=<dimen>
\extrarowdepth=<dimen>

```

`\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`.

```

\extrarowsep=<dimen>

```

`\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	$\alpha$ $\beta$	First $\frac{\partial \Phi}{\partial \theta}$
Three Four	Three Four	$\frac{\Phi}{\theta}$ $\Gamma_x^t$	Second $\frac{d\theta}{dt}$

`\extrarowsep=3mm`

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

`\extrarowsep=0mm`

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.

## 4.2 `\abovetabulinesep` `\belowtabulinesep` and `\tabulinesep`

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

`\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.

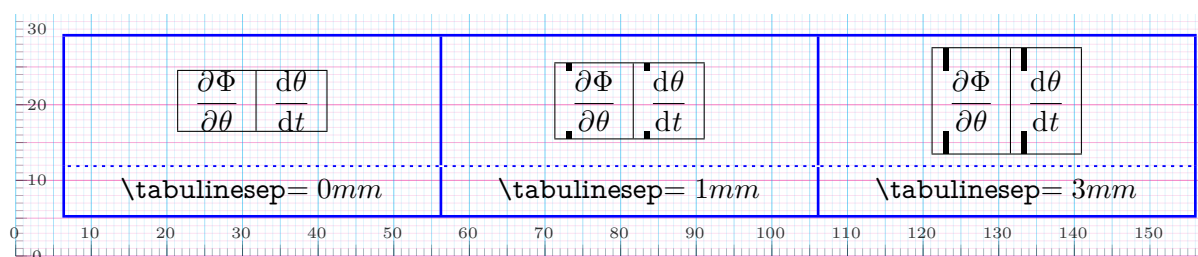
`\belowtabulinesep` is conversely the minimal vertical space allowed to the bottom border of the cell. When `\belowtabulinesep > 0pt` a vertical space is inserted – conditionally, depending on the cell content – below the cell content.

$$\backslash\text{tabulinesep}=\langle\textit{dimen}\rangle$$

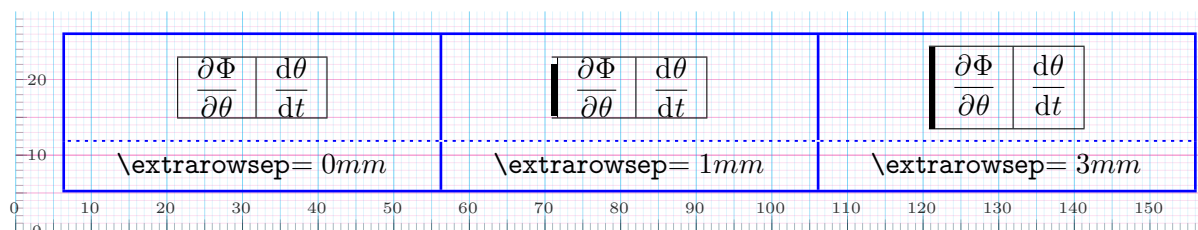
`\tabulinesep` makes the assignment for both `\abovetabulinesep` and `\belowtabulinesep`. The macro can be prefixed by `\global`<sup>5</sup>.

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

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



Now you can compare with `\extrarowsep`:



Presently (version 2.1), 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`.

$$\backslash\text{tabustrutrue}=\langle\textit{dimen}\rangle$$

For debugging purpose only (or to learn how this works): `\tabustrutrue` 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`.

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



## 5 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 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]-
\tabuphantomline
\end{tabu}
```

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

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`.

`\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`.

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

### 6.1 Paragraph indentation

See [Paragraph indentation](#)

### 6.2 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` `\preamble` (or `\usetabu`) for this purpose.

### 6.3 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 **cell content two one** |

→ 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`.

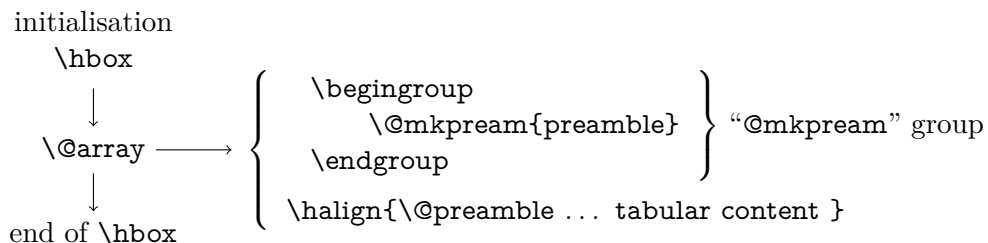
This rather strange inversion of tokens may be justified in math mode (otherwise, errors may occur) but not in text mode in our opinion. Inside a `tabu` environment, when not in math mode, the tokens are not reversed and you get the intuitively expected result:

| 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.

## 6.4 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:

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

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 recommended 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 !

## 7 The package options

### 7.1 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.

### 7.2 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\tabuledefaulttarget=\linewidth`

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

### 7.3 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.

### 7.4 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`, `\tabucline` and `\everyrow` 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.

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

### 8.1 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>

### 8.2 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.

## 9 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...



## 10 Implementation

### 10.1 Identification, requirements and options

The package namespace is `tabu@`.

```

1 \*package
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{tabu}[2011/01/19 v2.1 - 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
```

## 10.2 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@nbcols</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@nbcols 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@negcoef 100 \newif \iftabu@firstcline 101 \newif \iftabu@everyrow 102 \def\tabu@everyrowtrue{\global\let\iftabu@everyrow \iftrue} 103 \def\tabu@everyrowfalse{\global\let\iftabu@everyrow \iffalse}	
\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. 104 \newsavebox \tabu@box 105 \chardef \tabu@arstrutbox=\@arstrutbox 106 \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.	
\tabu@ifenvir	Check if the current environment is tabu or longtabu (for \multicolumnn). 107 \def\tabu@gobblespace#1 {#1}	

```

108 \def\tabu@gobbleopt#1{\@ifnextchar [{\tabu@gobble@pt{#1}}{#1}}%
109 \def\tabu@gobble@pt#1[#2]{#1}
110 \def\tabu@ifenvir {%
111   \csname @\expandafter\ifx\csname\@currentenv\endcsname\tabu
112   first\else\expandafter\ifx\csname\@currentenv\endcsname\longtabu
113   first\else second\fi\fi oftwo\endcsname
114 }% \tabu@ifenvir

```

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

```

115 \def\tabu@save@decl{%
116   \toks\count@ =\expandafter{\the\toks\expandafter\count@ \@nextchar}%
117 }% \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.

```

118 \def\tabu@newcolumnntype #1{%
119   \expandafter\tabu@new@columnntype
120   \csname NC@find@\string#1\expandafter\endcsname
121   \csname NC@rewrite@\string#1\endcsname
122   {#1}%
123 }% \tabu@newcolumnntype
124 \def\tabu@new@columnntype #1#2#3{%
125   \def#1##1#3{\NC@{##1}}%
126   \let#2\relax \newcommand*#2%
127 }% \tabu@new@columnntype
128 \def\tabu@privatecolumnntype #1{%
129   \expandafter\tabu@private@columnntype
130   \csname NC@find@\string#1\expandafter\endcsname
131   \csname NC@rewrite@\string#1\expandafter\endcsname
132   \csname tabu@NC@find@\string#1\expandafter\endcsname
133   \csname tabu@NC@rewrite@\string#1\endcsname
134   {#1}%
135 }% \tabu@privatecolumnntype
136 \def\tabu@private@columnntype#1#2#3#4{%
137   \g@addto@macro\tabu@privatecolumns{\let#1#3\let#2#4}%
138   \tabu@new@columnntype#3#4%
139 }% \tabu@private@columnntype
140 \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}`

```

141 \tabu@privatecolumnntype X[1][]{\tabu@rewrite@X{#1}}%
142   \let\@halignto \relax
143   \expandafter \NC@find \tabucolX\rw}

```

The next part of the definition (`\tabu@rewrite@X`) can be found page 34.

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

```

144 \tabu@privatecolumnntype \usetabu [1]{%
145   \ifx\\#1\\\tabu@saveerr{}}\else
146   \ifundefined\tabu@saved@\string#1}
147   {\tabu@saveerr{#1}}
148   {\let\tabu@rewrite@X \tabu@rewrite@Xrestore

```

```

149         \csname tabu@saved@\string#1\expandafter\endcsname\expandafter\z@}%
150     \fi
151 }% \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.

```

152 \tabu@privatecolumnntype \preamble [1]{%
153     \ifx\#1\\\tabu@saveerr{}\else
154         \ifundefined{tabu@saved@\string#1}
155             {\tabu@saveerr{#1}}
156             {\csname tabu@saved@\string#1\expandafter\endcsname\expandafter\@ne}%
157     \fi
158 }% \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).

```

159 \tabu@newcolumnntype \tabu@rewritefirst{%
160     \tabu@X@cols \z@
161     \let\tabu@dimexpr \relax
162     \let\tabu@savepreamble \relax
163     \let\NC@rewrite@X \tabu@NC@rewrite@X
164     \gdef\tabu@global@temp {\tabu@savepreamble}%
165     \aftergroup \tabu@global@temp
166     \let\tabu@islast \relax
167     \tabu@privatecolumns
168     \NC@list{\NC@do\usetabu \NC@do\preamble \NC@do *}%
169     \tabu@lines % defines NC@rewrite@ for tabu only (inside @mkpream group)
170     \NC@list\expandafter{\the\expandafter\NC@list \expandafter\NC@do \expandafter X%
171         \tabu@NC@list \NC@do X\NC@do\tabu@rewritemiddle}%
172     \ifcat$\dollar\end\else
173         \let\save@decl \tabu@save@decl \fi % no inversion of tokens in text mode
174     \NC@find \tabu@rewritemiddle
175 }% \NC@rewrite@\tabu@rewritefirst
176 \def\tabu@savepreamble{%
177     \NC@list\expandafter{\tabu@NC@list}%
178     \let\tabu@savedpreamble \@preamble
179     \ifdim \tabustrutrule>\z@
180         \setbox\@arstrutbox\hbox{\vrule
181             \@height \ht\@arstrutbox
182             \@width \tabustrutrule
183             \@depth \dp\@arstrutbox}%
184     \fi
185     \ifdim \extrarowdepth>\z@
186         \setbox\@arstrutbox\hbox{\vrule
187             \@height \ht\@arstrutbox
188             \@width \tabustrutrule
189             \@depth \dimexpr\dp\@arstrutbox+\arraystretch\extrarowdepth}%
190     \fi
191 }% \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.

```

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

```



```

239 }% \tabu@extracol@sep
240 \def\tabu@global@Xnested{\expandafter\tabu@global@X@nested
241     \csname tabu@global@X\the\tabu@nested\endcsname}
242 \def\tabu@global@X@nested#1{%
243     \global\let#1\tabu@global@X
244     \aftergroup\let \aftergroup\tabu@global@X \aftergroup#1%
245 }% \tabu@global@Xnested

```

**\tabu@rewrite@multicolumn** A special rewrite to allow |...| in \multicolumn preamble inside tabu environment.

```

246 \tabu@newcolumnntype \tabu@rewrite@multicolumn{%
247     \aftergroup\endgroup % after \@mkpream group (restores \multispan)
248     \tabu@lines \NC@list\expandafter{\the\expandafter\NC@list \tabu@NC@list}%
249     \ifcat$\dollar\end\else
250         \let\save@decl \tabu@save@decl \fi % no inversion of tokens in text mode
251     \NC@find
252 }% \tabu@rewrite@multicolumn
253 \def\tabu@multicolumn{\tabu@ifenvir \tabu@multic@lumn \tabu@multicolumnORI}
254 \def\tabu@multic@lumn#1#2{\multispan{#1}\begingroup
255     \NC@list{\NC@do \tabu@rewrite@multicolumn}%
256     \expandafter\@gobbletwo % gobbles \multispan{#1}
257     \tabu@multicolumnORI{#1}{\tabu@rewrite@multicolumn #2}%
258 }% \tabu@multic@lumn

```

**\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.

```
259 \newif\iftabu@colortbl
```

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

```

260 \def\tabu@nowrite#1#{{\afterassignment}\toks@}
261 \let\tabu@write\write
262 \let\tabu@immediate\immediate
263 \protected\def\tabu@GenericError{\begingroup
264     \def\immediate\write{\aftergroup\endgroup
265         \tabu@immediate\tabu@write}\tabu@GenericErrorORI}%
266 \let\tabu@GenericErrorORI \GenericError
267 \protected\def\tabu@warn{\begingroup
268     \def\immediate\write{\aftergroup\endgroup
269         \tabu@immediate\tabu@write}\PackageWarning\tabu}

```

## 10.3 \tabu, \endtabu, \longtabu and \endlontabu

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

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

```

270 \def\tabu{%
271     \ifmmode \def\tabu@{\array}\def\endtabu{\endarray}%
272     \else \def\tabu@{\tabular}\def\endtabu{\endtabular}\fi
273     \let\tabu@arrayleft@measure \tabu@arrayleftmeasure
274     \tabu@setup \tabu@settarget
275 }% \tabu
276 \def\longtabu{%
277     \ifmmode\PackageError\tabu{longtabu not allowed in math mode}\fi
278     \def\tabu@{\longtable}\def\endlongtabu{\endlongtable}%
279     \let\tabu@arrayleft@measure \tabu@long@arrayleft@measure
280     \let\LT@startpbox \tabu@LT@startpbox % \everypar{ array strut } instead of { array strut }
281     \tabu@setup \tabu@settarget
282 }% \longtabu

```

```

283 \def\tabu@nolongtabu{\PackageError{tabu}
284   {longtabu requires the longtable package}\@ehd}

```

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

```

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

```

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

```

\tabu@begin
337 \def\tabu@settarget{\futurelet\@let@token \tabu@sett@rget}
338 \def\tabu@sett@rget{\tabu@target\z@
339   \ifcase \ifx \bgroup\@let@token 0\else
340     \ifx [\@let@token 0\else
341     \ifx \@sptoken\@let@token 1\else
342     \if t\@let@token 2\else
343     \if s\@let@token 3\else
344     \m@ne\fi\fi\fi\fi\fi\relax
345   \expandafter\tabu@begin
346   \or \expandafter\tabu@gobblespace\expandafter\tabu@settarget
347   \or \expandafter\tabu@to
348   \or \expandafter\tabu@spread
349   \else \expandafter\tabu@begin
350   \fi
351 }% \tabu@sett@rget
352 \def\tabu@to to{\def\tabu@halignto{to}\tabu@gettarget}
353 \def\tabu@spread spread{\tabu@spreadtrue\def\tabu@halignto{spread}\tabu@gettarget}
354 \def\tabu@gettarget {\afterassignment\tabu@linegoaltarget \tabu@target}
355 \def\tabu@linegoaltarget {\futurelet\tabu@temp \tabu@linegoal@target}
356 \def\tabu@linegoal@target {%
357   \ifx \tabu@temp\LNGl@setlinegoal
358     \LNGl@setlinegoal \expandafter \@firstoftwo \fi % @gobbles \LNGl@setlinegoal
359   \tabu@begin
360 }% \tabu@linegoal@target
361 \def\tabu@begin #1#{%
362   \edef\tabu@halignto{\ifdim\tabu@target>\z@ \tabu@halignto\the\tabu@target\fi}%
363   \@testopt \tabu@@begin \tabu@aligndefault #1\@nil
364 }% \tabu@begin
365 \def\tabu@@begin [#1]#2\@nil#3{%
366   \tabu@preparesave{#3}% <for \savetabu>
367   \def\tabu@align {#1}%
368   \tabu@ [\tabu@align]#2{\tabu@rewritefirst #3}%
369 }% \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.

```

370 \def\tabu@preparesave #1{%
371   \iftabu@colortbl \@temptokena{}%
372   \tabu@prepare@save \CT@arc@
373   \tabu@prepare@save \CT@drsc@
374   \fi
375   \edef\tabu@sav@param{%
376     \ifmmode \col@sep \the\arraycolsep
377     \else \col@sep \the\tabcolsep \fi\relax
378     \arrayrulewidth \the\arrayrulewidth\relax
379     \doublerulesep \the\doublerulesep\relax
380     \extrarowheight \the\extrarowheight\relax
381     \extrarowdepth \the\extrarowdepth\relax
382     \extratabsurround \the\extratabsurround\relax
383     \def\noexpand\arraystretch {\arraystretch}%
384     \iftabu@colortbl \minrowclearance\the\minrowclearance\relax
385     \the\@temptokena
386   \fi}%
387   \def\tabu@sav@pream {\edef\tabu@halignto{to\the\tabu@target}\NC@find #1}%
388 }% \tabu@preparesave

```

```

389 \def\tabu@prepare@save #1{%
390   \ifx\relax#1\else
391     \@temptokena\expandafter{\the\expandafter\@temptokena
392       \expandafter\def\expandafter#1\expandafter{#1}}\fi
393 }% \tabu@prepare@save

```

## 10.4 Footnotes inside tabu

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

```
394 \newtoks\tabu@footnotes
```

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

```

395 \long\def\tabu@footnotetext #1{%
396   \edef\@tempa{\the\tabu@footnotes
397     \noexpand\footnotetext [\the\csname c@\@mpfn\endcsname]}%
398   \global\tabu@footnotes\expandafter{\@tempa {#1}}}%
399 \long\def\tabu@xfootnotetext [#1]#2{%
400   \global\tabu@footnotes\expandafter{\the\tabu@footnotes
401     \footnotetext [{#1}]{#2}}}%
402 \let\tabu@xfootnote \@xfootnote

```

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

`\tabu@Hy@xfootnote`

```

403 \long\def\tabu@Hy@ftntext{\tabu@Hy@ftntxt {\the \c@footnote }}
404 \long\def\tabu@Hy@xfootnote [#1] {%
405   \begingroup
406     \value\@mpfn #1\relax
407     \protected@xdef \@thefnmark {\thempfn}%
408   \endgroup
409   \@footnotemark \tabu@Hy@ftntxt {#1}%
410 }% \tabu@Hy@xfootnote
411 \long\def\tabu@Hy@ftntxt #1#2{%
412   \edef\@tempa{%
413     \the\tabu@footnotes
414     \begingroup
415       \value\@mpfn #1\relax
416       \noexpand\protected@xdef
417         \noexpand\@thefnmark{\noexpand\thempfn}%
418       \expandafter\noexpand\expandafter\tabu@Hy@footnotetext
419       \expandafter{%
420         \Hy@footnote@currentHref
421       }%
422     }%
423     \global\tabu@footnotes\expandafter{%
424       \@tempa {#2}}%
425     \endgroup
426   }%
427 }% \tabu@Hy@ftntxt
428 \long\def\tabu@Hy@footnotetext #1#2{%
429   \H@@footnotetext{%
430     \ifHy@nesting
431       \hyper@@anchor {#1}{#2}%
432     \else
433       \Hy@raisedlink{%
434         \hyper@@anchor {#1}{\relax}%
435       }%
436       \def\@currentHref {#1}%
437       \let\@currentlabelname \@empty
438     #2%

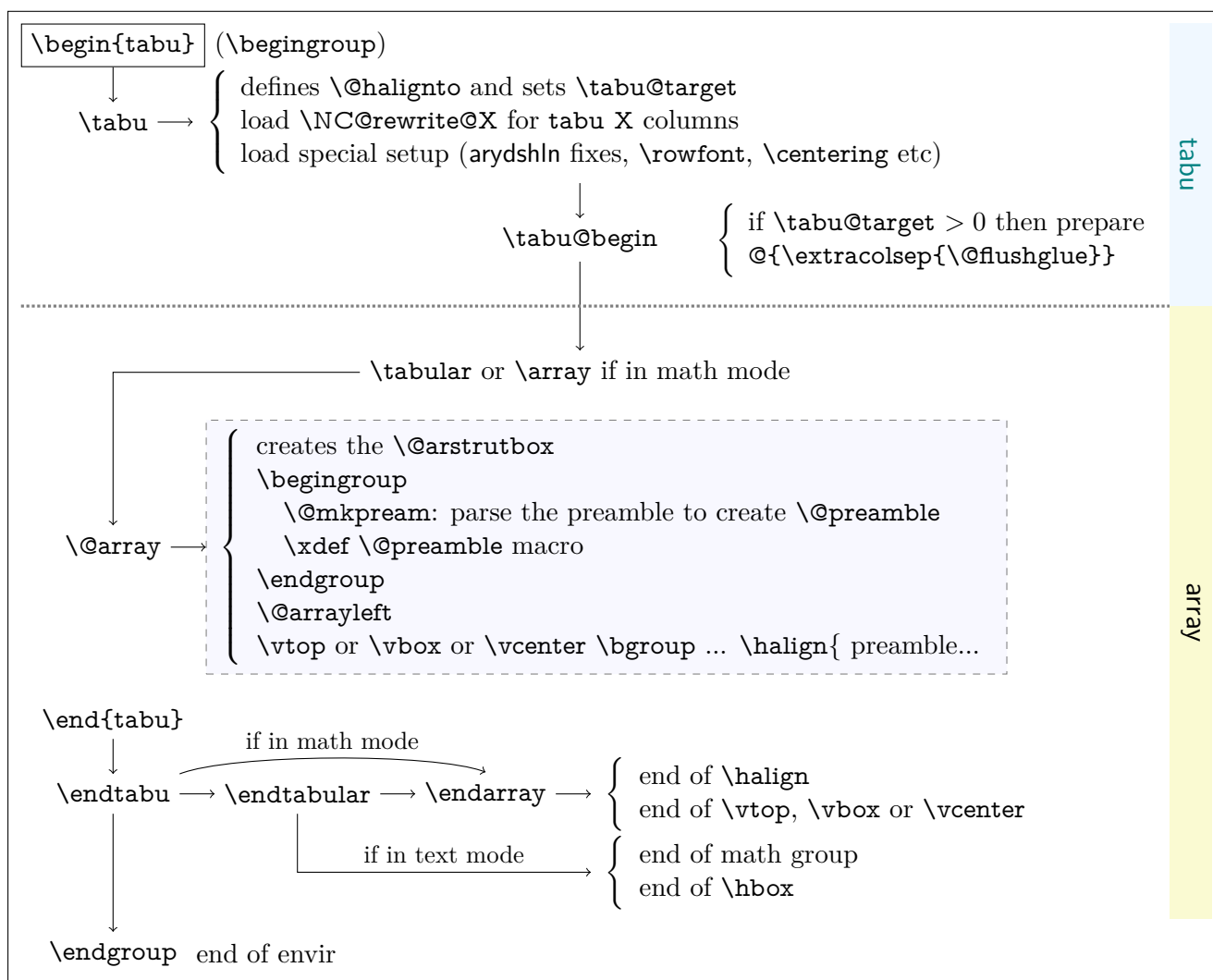
```

```

439     \fi
440   }%
441 }% \tabu@Hy@footnotetext
442 \AtBeginDocument{%
443   \csname\ifcsname ifHy@hyperfootnotes\endcsname
444     ifHy@hyperfootnotes\else iffalse\fi\endcsname
445     \let\tabu@footnotetext \tabu@Hy@ftntext
446     \let\tabu@xfootnote   \tabu@Hy@xfootnote
447   \fi
448 }
```

## 10.5 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`.

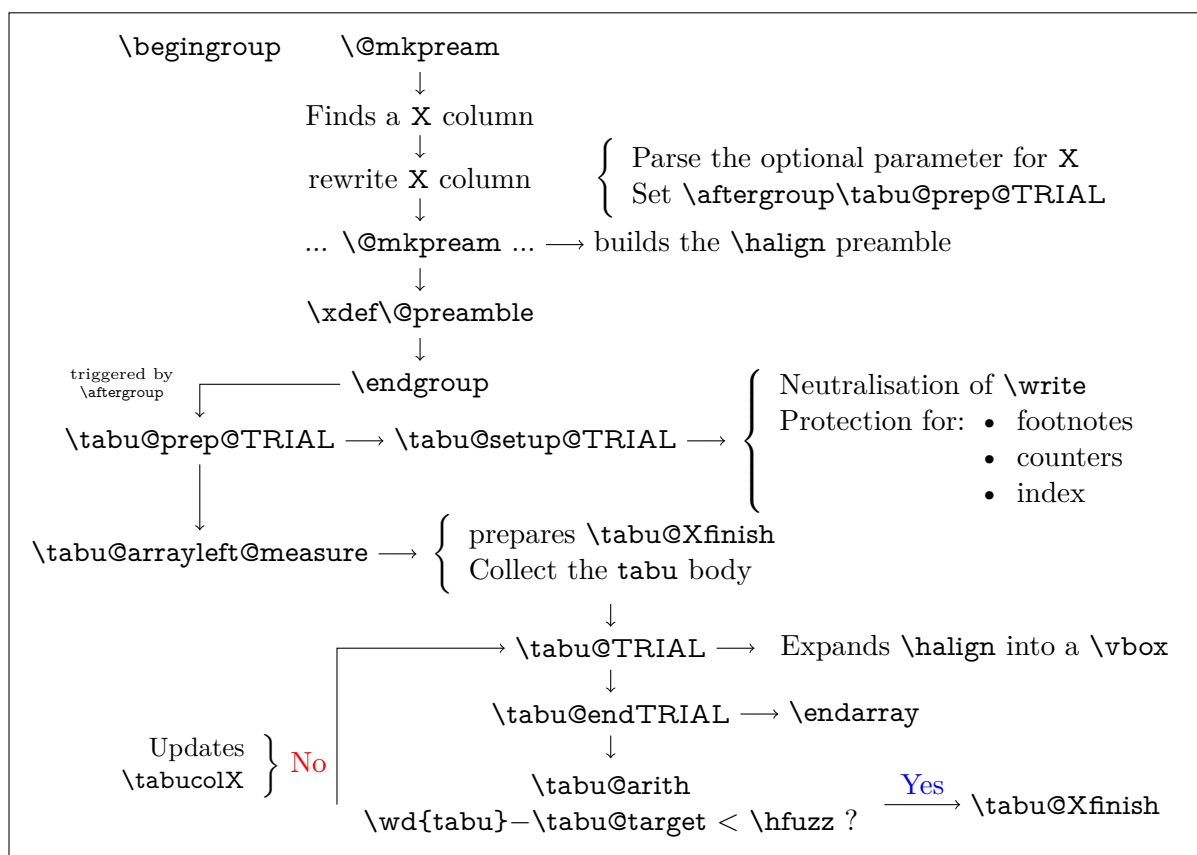
When a `tabu X` column is found in the preamble by `\@mkpreamble`, `tabu` changes his strategy: the macro `\tabu@prep@TRIAL` is set to be expanded `\aftergroup` that is, just after the preamble (`\@preamble`) has been built. This macro does some setup for `tabu` trials to reach the target with variable `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.

## 10.6 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.
- `\@halignto` is `\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`.

```

449 \def\tabu@rewrite@X #1{\tabu@Xarg {#1}%
450   \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \relax
451   \iftabu@measuring % not the first X column found in preamble
452     \edef\tabu@global@X {\tabu@global@X
453       \def\csname tabu@X\the\tabu@X@cols\endcsname {\strip@pt\tabu@X@sum}%
454       \advance\tabu@X@sum \the\tabu@X@sum\relax}%
455   \else % first X column found in preamble
456     \tabu@measuringtrue
457     \iftabu@spread \tabu@spreadtarget\tabu@target \tabu@target\z@
458     \else          \tabu@spreadtarget \z@ \fi
459     \ifdim\tabu@target=\z@
460       \setlength\tabu@target \tabu@defaulttarget
461       \tabu@message{(tabu) Default target:
462         \ifx\tabu@defaulttarget\linewidth\string\linewidth
463         \else\ifx\tabu@defaulttarget\linegoal\string\linegoal
464         \fi\fi=\the\tabu@target \on@line, page \the\c@page}\fi
465     \tabu@message{\tabu@titles}%
466     \edef\tabu@global@X {\tabu@global@X
467       \def\csname tabu@X1\endcsname{\strip@pt\tabu@X@sum}%
468       \tabu@X@sum \the\tabu@X@sum\relax
469       \tabu@target \the\tabu@target\relax
470       \tabu@spreadtarget \the\tabu@spreadtarget\relax}%
471     \let\tabu@halignto \relax
472   \fi
473 }% \tabu@rewrite@X
474 \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`.

```

475 \def\tabu@rewrite@Xnested #1{\in@ ${#1}\ifin@
476   \def\tabucolX@rw {>{${#1}}p{\tabucolX}<{${#1}}}\else
477   \def\tabucolX@rw {p{\tabucolX}}\fi}
478 \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`

```

479 \def\tabu@Xarg#1{%
480   \advance\tabu@X@cols \@ne
481   \tabu@wdlet\tabu@X@cols \z@
482   \ifx\#1\% <shortcut when no option>
483     \tabu@X@sum \p@

```

```

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

```

```

542     {Duplicate vertical alignment specification}\fi
543     \def\tabucolX@rw{#1}\expandafter\tabu@Xparse
544 }% \tabu@Xtype
545 \def\tabu@Xcoef#1{\edef\tabu@temp{\tabu@temp#1}%
546   \afterassignment\tabu@Xc@ef \tabu@cnt\number\if-#10\fi
547 }% \tabu@Xcoef
548 \def\tabu@Xc@ef{%
549   \advance\tabu@X@sum \tabu@temp\the\tabu@cnt\p@
550   \tabu@Xparse}%
551 }% \tabu@Xc@ef
552 \def\tabucolX@do{%
553   \ifx\tabucolX@rw\@empty \def\tabucolX@rw{p}\fi
554   \ifdim\tabu@X@sum<\z@ \tabu@negcoeftrue
555     \tabu@X@sum \m@ne\tabu@X@sum
556     \tabu@wdlet \tabu@X@cols \p@
557   \else \ifdim \tabu@X@sum=\z@ \tabu@X@sum \p@ \fi
558   \fi
559   \begingroup
560     \toks@{\expandafter {\tabucolX@align}}%
561     \edef\tabu@temp {\strip@pt\tabu@X@sum \tabucolX}%
562     \edef\tabucolX@rw##1{%
563       \ifx$\tabu@Xmath>\{\noexpand\the\toks@ $\}\else
564       \ifx \tabucolX@align\@empty \else
565         >\{\noexpand\the\toks@\}\fi\fi
566       \tabucolX@rw{##1}%
567       \ifx$\tabu@Xmath<\{$\}\fi}%
568     \edef\x{\endgroup
569       \def\noexpand\tabu@temp {\tabucolX@rw{\the\dimexpr\tabu@temp}}%
570       \def\noexpand\tabucolX@rw{%
571         \tabucolX@rw{\tabu@dimexpr{\the\tabu@X@cols}{\tabu@temp}}}%
572     }\x
573 }% \tabucolX@do
574 \def\tabu@dimexpr#1#2{%
575   \ifdim \tabu@wd{#1}>\p@
576     \ifdim \tabu@wd{#1}<#2 \tabu@wd{#1}\else#2\fi\else #2\fi
577 }% \tabu@dimexpr

```

## 10.7 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`.

```

578 \def\tabu@prep@TRIAL{%
579   \begingroup \iffalse{\fi \ifnum0='}\fi
580   \tabu@setup@TRIAL
581   \tabucolX \tabu@target
582   \ifcase 0\iftabu@negcoef \else\iftabu@spread \else 1\fi\fi
583     \advance\tabu@X@cols \@ne
584     \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \@undefined
585     \tabu@X@cols \z@
586     \def\tabu@naturalX@max {\z@}%
587     \def\tabu@naturalX@min {\z@}%
588     \let\tabu@startpboxORI \@startpbox
589     \let\@startpbox \tabu@measure@startpbox
590   \fi

```

```

591 \def\tabu@lasttry{\m@ne\p}\tabu@cnt \z@ % number of trials
592 \tabu@arrayleft@measure
593 }% \tabu@prep@TRIAL

\tabu@setup@TRIAL

594 \def\tabu@setup@TRIAL{\tabu@global@X \let\tabu@global@X \@empty
595 \let\@halignto \@empty \tabu@naturalX \z@
596 \tabu@savecounters \let\tabu@savecounters \relax
597 \hbadness \@M \let\hbadness \@tempcnta
598 \hfuzz \maxdimen \let\hfuzz \@tempdima
599 \let\tabu@rewrite@X \tabu@rewrite@Xnested
600 \let\savetabu \@gobble \let\everyrow \@gobble
601 \let\tabudefaulttarget \linewidth
602 \let\write \tabu@nowrite \let\GenericError \tabu@GenericError
603 \long\def\@xfootnote[##1]##2{}\let\@footnotetext \@gobble
604 \ifx\@arrayright\relax \let\@arrayright \@empty\fi
605 \let\tabu@verticalmeasure \@empty
606 \let\tabu@verticalspacing \@empty
607 \tabu@TRIAL@hook
608 }% \tabu@setup@TRIAL
609 \def\tabu@savecounters{%
610 \def\@elt##1{\global\value{##1}\the\value{##1}\relax}%
611 \edef\tabu@global@X {\cl@ckpt}\let\@elt \relax
612 }% \tabu@savecounters

\tabuDisableCommands

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

```

## 10.8 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`.

```

615 \def\tabu@arrayleftmeasure#1\bggroup{%
616 \def\tabu@Xfinish{%
617 \tabu@global@Xnested \ifnum0='{}\fi\endgroup
618 \ifdim \abovetabulinesep>\z@ \setbox\@arstrutbox\box\voidb@x
619 \else\ifdim \belowtabulinesep>\z@ \setbox\@arstrutbox\box\voidb@x
620 \fi\fi
621 \tabu@global@X #1\bggroup}%
622 \def\tabu@TRIAL {\setbox\tabu@box \hbox\bgroup $#1\bgroup
623 \the\toks@ \tabu@endTRIAL}%
624 \toks@{\tabu@setup@everycr
625 \let\@preamble \tabu@savedpreamble}% <required for multicolumn>
626 \tabu@collect
627 }% \tabu@arrayleftmeasure

```

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

```

628 \long\def\tabu@long@arrayleft@measure #1\LT@bchunk #2\LT@bchunk{%
629 \def\tabu@Xfinish{%
630 \tabu@global@Xnested \ifnum0='{}\fi\endgroup
631 \tabu@setup@everycr \let\@preamble \tabu@savedpreamble
632 \tabu@global@X #1\LT@bchunk #2\LT@bchunk}%
633 \def\tabu@TRIAL{%
634 \setbox\tabu@box \hbox\bgroup
635 \tabu@setup@everycr \let\@preamble \tabu@savedpreamble

```

```

636 \tabu@global@X \LTchunksize\@M
637 #1\LT@bchunk #2\LT@bchunk \the\toks@ \tabu@long@endTRIAL}%
638 \toks@{}\let\LT@output\relax
639 \tabu@collect
640 }% \tabu@long@arrayleft@measure
641 \def\tabu@long@endTRIAL{%
642 \LT@echunk
643 \global\setbox\@ne \hbox{\unhbox\@ne}\kern\wd\@ne
644 \LT@get@widths
645 \egroup % got \tabu@box
646 \iftabu@spread \tabu@spreadarith % <only once>
647 \else \tabu@arith
648 \fi
649 \iftabu@measuring \tabu@measuringfalse
650 \expandafter \tabu@TRIAL % <continue trials>
651 \else
652 \expandafter \tabu@Xfinish
653 \the \expandafter\toks@ \tabu@endenvir % <we are then!>
654 \fi
655 }% \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`.

```

656 \def\tabu@collect{\catcode'\^@=13\def\tabu@stack{b}\tabu@collectbody}
657 \long\def\tabu@collectbody#1\end#2{%
658 \edef\tabu@stack{\tabu@pushbegins #1\begin\end\expandafter\@gobble\tabu@stack}%
659 \ifx \tabu@stack\@empty
660 \toks@\expandafter{\the\toks@#1}\def\tabu@endenvir{\end{#2}}%
661 \expandafter \tabu@TRIAL
662 \else
663 \toks@\expandafter{\the\toks@#1\end{#2}}%
664 \expandafter \tabu@collectbody
665 \fi
666 }% \tabu@collectbody
667 \long\def\tabu@pushbegins#1\begin#2{%
668 \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`.

```

669 \def\tabu@endTRIAL{%
670 \tabu@adl@endTRIAL % <arydshln in nested trials>
671 \endarray$\egroup
672 \iftabu@spread \tabu@spreadarith % <only once>
673 \else \tabu@arith
674 \fi
675 \iftabu@measuring \tabu@measuringfalse
676 \expandafter \tabu@TRIAL % <continue trials>
677 \else
678 \expandafter \tabu@Xfinish
679 \the \expandafter\toks@ \tabu@endenvir % <we are then!>
680 \fi
681 }% \tabu@endTRIAL
682 \def\tabu@adl@endTRIAL{\crcr \noalign{\global\adl@nocol \tabu@nbcols}}

```



## 10.9 Algorithms: Measuring the tabu

### The arithmetic of X columns

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

```

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

```



```

736         \ifdim \tabu@wd\tabu@X@cols <\tabu@temp\tabucolX
737             \advance\tabu@X@sum -\tabu@temp\p@
738             \tabu@message{\tabu@message@negcoef{\the\tabu@X@cols}}}%
739             \ifdim \tabu@wd\tabu@X@cols >\tabu@wd\count@ \count@ \tabu@X@cols \fi
740         \fi\fi
741         \advance\tabu@X@cols \@ne
742     }%
743     \@tempdima=\dimexpr\tabu@wd\count@ * \p@/
744             \dimexpr\csname tabu@X\the\count@\endcsname\p@\relax\relax
745 }% \tabu@arith@negcoef

```

## The arithmetic of X columns for tabu spread

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

```

746 \def\tabu@spreadarith{%
747     \tabu@naturalX \tabu@naturalX@max\relax
748     \tabucolX \tabu@naturalX@min\relax
749     \tabu@message{\tabu@message@spread}%
750     \ifdim \dimexpr \wd\tabu@box + \tabu@spreadtarget < \tabu@target
751         \advance\tabu@spreadtarget
752             \dimexpr \wd\tabu@box - \tabu@naturalX + \tabucolX \relax
753     \ifdim \tabu@spreadtarget < \tabu@target
754         \tabu@target = \tabu@spreadtarget
755     \fi
756     \tabu@message{\the\tabu@target}%
757     \tabu@message{\ifdim \tabu@spreadtarget > \tabu@target
758         (tabu) spread reduced to fit default target.\fi}%
759 \else
760     \tabu@message{\expandafter\tabu@msg@align \the\tabu@target { }{}{}{}{}@}%
761     \tabu@message{(tabu)\ifdim \wd\tabu@box < \tabu@target spread too large\else
762         tabu spread is useless here\fi,
763         default target used.}%
764 \fi
765 \let\@startpbox \tabu@startpboxORI
766 \xdef\tabu@halignto {to\the\tabu@target}%
767 \tabucolX \tabu@target
768 \tabu@spreadfalse \tabu@measuringtrue
769 }% \tabu@spreadarith

```

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

```

\tabu@message@arith
\tabu@message@spread
\tabu@message@negcoef
770 \def\tabu@message@arith{\tabu@header
771   \expandafter\tabu@msg@align \the\tabucolX { }{ }{ }{ }{\}\@@
772   \expandafter\tabu@msg@align \the\wd\tabu@box { }{ }{ }{ }{\}\@@
773   \expandafter\tabu@msg@align \the\tabu@target { }{ }{ }{ }{\}\@@
774   \expandafter\tabu@msg@align \the\tabu@X@sum {}{}{}{}{}{\}\@@
775   \ifx\@startpbox\tabu@measure@startpbox\tabu@spaces (coefs<0)\else
776   \ifdim\tabu@X@sum>\z@
777   \expandafter\tabu@msg@align \the\dimexpr(\tabu@target-\wd\tabu@box)
778   *p@/\tabu@X@sum\relax {}{}{}{}{}{\}\@@\fi\fi^^J}%
779 \def\tabu@message@spread{\tabu@spreadheader
780   \expandafter\tabu@msg@align \the\tabu@spreadtarget {}{}{}{}{}{\}\@@
781   \expandafter\tabu@msg@align \the\wd\tabu@box {}{}{}{}{}{\}\@@
782   \expandafter\tabu@msg@align \the\tabucolX {}{}{}{}{}{\}\@@
783   \expandafter\tabu@msg@align \expandafter-\the\tabu@naturalX {}{}{}{}{}{\}\@@
784 \def\tabu@message@reached{\tabu@header
785   \expandafter\tabu@msg@align \the\tabucolX { }{ }{ }{ }{\}\@@

```

```

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

```

```

844     \endgroup}%
845 }\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).

```

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

```

```

889 \ifdim \tabu@wd\tabu@X@cols>\z@
890 \ifdim \tabu@wd\tabu@X@cols<\wd\tabu@box
891 \tabu@wddef\tabu@X@cols {\the\wd\tabu@box}%
892 \tabu@debug{\tabu@spaces\tabu@spaces
893 * \the\tabu@X@cols. X[-\csname tabu@X\the\tabu@X@cols\endcsname]:
894 target=\the\dimexpr\tabu@temp\relax\space
895 \expandafter\expandafter\expandafter\string\tabu@wd\tabu@X@cols
896 =\tabu@wd\tabu@X@cols}%
897 \fi\fi
898 \box\tabu@box \egroup % end of \vtop (measure)
899 }% \tabu@measure@endpbox
900 \def\tabu@wd#1{\csname tabu@W\the\tabu@nested @\number#1\endcsname}
901 \def\tabu@wddef#1{\expandafter\xdef
902 \csname tabu@W\the\tabu@nested @\number#1\endcsname}
903 \def\tabu@wdlet#1{\global\expandafter\let
904 \csname tabu@W\the\tabu@nested @\number#1\endcsname}
905 \def\tabu@naturalX@min{\z@}
906 \def\tabu@naturalX@max{\z@}

```

## 10.10 \everycr setup and \tabuphantomline

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

```

907 \def\tabu@setup@everycr{%
908 \tabu@firstclinettrue \tabu@everyrowtrue
909 \let\tabu@first@ccline \z@
910 \def\tabu@restoreeverycr {\let\everycr \tabu@everycr}%
911 \def\everycr{\afterassignment\tabu@restoreeverycr \@temptokena}% <for ialign>
912 \tabu@everycr\expandafter{\the\tabu@everycr \tabu@everyrow}%
913 }% \tabu@setup@everycr
914 \def\tabu@everyrow{%
915 \noalign{%
916 \ifcase\tabu@first@ccline
917 \global\let\tabu@first@ccline \tw@
918 \or\global\let\tabu@first@ccline \tw@
919 \or\global\tabu@firstcclinefalse
920 \global\let\tabu@first@ccline \m@ne
921 \fi}%
922 \tabu@rowfontreset
923 \iftabu@everyrow
924 \noalign{\tabu@everyrowfalse}\expandafter\tabu@everyrow@hook
925 \fi
926 }% \tabu@everyrow
927 \let\tabu@everyrow@hook \@empty

```

`\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...`

```

928 \def\tabuphantomline{\crrc\noalign{%
929 \global\let\@arstrutbox \voidb@x
930 \iftabu@colortbl
931 \xdef\tabu@minrowclearance{\the\minrowclearance}%
932 \global\minrowclearance \z@
933 \fi
934 \global\let\vcenter \vbox
935 \ifnum \tabu@nested>\z@ \vspace*{\dimexpr\extrarowheight-\ht\@arstrutbox}\fi
936 \toks@{}\count@\@ne
937 \@whilenum\count@<\tabu@nbc\do{\advance\count@\@ne
938 \toks@\expandafter{\the\toks@&}}%

```

```

939 \toks@{\expandafter{\the\toks@
940 \cr\noalign{%
941 \global\let\@arstrutbox \tabu@arstrutbox
942 \global\let\vcnter \tabu@vcenter
943 \iftabu@colortbl
944 \global\minrowclearance \tabu@minrowclearance\relax\fi
945 \global\let\tabu@first@cline \@ne}}%
946 \expandafter}\the\toks@
947 }% \tabuphantomline

```

## 10.11 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.

```

948 \def\tabu@lines{{\endlinechar\m@ne \scantokens{\def\x{}}}%
949 \expandafter}\expandafter\tabu@lines@ \x
950 }% \tabu@lines
951 \def\tabu@lines@#1{%
952 \tabu@newcolumnntype #1{\tabuvline@rewrite}%
953 \NC@list\expandafter{\the\NC@list \NC@do #1}%
954 }% \tabu@lines@
955 \newcommand*\tabuvline@rewrite[1][\tabuvline@arg{#1}%
956 \expandafter\NC@find \tabu@temp}
957 \def\tabu@vline#1{\vrule width#1}

```

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

```

958 \def\tabuvline@arg#1{%
959 \ifx\#1\% <shortcut when no option>
960 \edef\tabu@temp{!{%
961 \iftabu@colortbl\begin{group}\noexpand\CT@arc@{\fi
962 \noexpand\tabu@vline\arrayrulewidth
963 \iftabu@colortbl\end{group}\fi}}%
964 \else
965 \ifcename tabu@line@style@\detokenize {#1}\endcename
966 \def\tabu@temp##1\relax##2\relax{##1}%
967 \expandafter\tabu@temp\romannumeral-'\q
968 \cename tabu@line@style@\detokenize {#1}\endcename
969 \tabuvline@argiii
970 \else \futurelet \tabu@temp \tabuvline@argi #1\p@\p@\@nnil \tabuvline@argiii
971 \fi
972 \fi
973 }% \tabuvline@arg
974 \def\tabuvline@argi{%
975 \let\tabu@color \@empty
976 \ifcat A\noexpand\tabu@temp
977 \@tempdima\arrayrulewidth
978 \expandafter\tabu@getlinecolor
979 \else \expandafter\tabuvline@argii
980 \fi
981 }% \tabuvline@argi
982 \def\tabuvline@argii#1\@nnil{%
983 \tabu@maybecolor \@tempdima #1 \@nnil
984 }% \tabuvline@argii
985 \def\tabuvline@argiii{%

```

```

986 \edef\tabu@temp{!\begingroup
987 \ifx\tabu@color\@empty
988 \iftabu@colortbl \noexpand\CT@arc@ \fi
989 \else \tabu@color\fi
990 \noexpand\tabu@vline{\the\@tempdima}\endgroup}}%
991 }% \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.

```

992 \def\tabu@linedash{4pt}% <default value>
993 \def\tabu@dashgap{4pt}% <default value>
994 \def\tabucline{\crcr\noalign{\ifnum0='}\fi \@ifstar
995 {\@tempswatrue \tabu@ccline}
996 {\@tempswafalse \tabu@ccline}}%
997 \newcommand*\tabu@ccline[2][\toks@]{%
998 \tabu@startstop{#2}\tabu@getlinespec{#1}\@multicnt\@ne
999 \ifnum\tabu@start>\tabu@stop
1000 \else\ifx\tabu@xleaders\relax
1001 \else
1002 \toks@{\noalign{\global\let\@arstrutbox \voidb@x}}%
1003 %% \iftabu@firstccline\if\tabu@align t\vskip-\ht\@arstrutbox\fi\fi
1004 \@whilenum\@multicnt<\tabu@start\do{\advance\@multicnt\@ne
1005 \toks@\expandafter{\the\toks@ &}}%
1006 \loop
1007 \toks@\expandafter{\the\toks@ \omit }%
1008 \ifnum \@multicnt=\@ne
1009 \expandafter \tabu@add \csname tabu@tok@\the\@multicnt L\endcsname \@nil
1010 \fi
1011 \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
1012 \toks@\expandafter{\the\toks@ \begingroup}%
1013 \toks@\expandafter{\the\expandafter\toks@
1014 \tabu@xleaders \endgroup}%
1015 \fi
1016 \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
1017 \expandafter \tabu@add \csname tabu@tok@\the\@multicnt R\endcsname \@nil
1018 \fi
1019 \ifnum\@multicnt<\tabu@stop
1020 \advance\@multicnt\@ne
1021 \if@tempswa \toks@\expandafter{\the\toks@ &}%
1022 \else \toks@\expandafter{\the\toks@ \span}\fi
1023 \repeat
1024 \@whilenum\@multicnt<\tabu@ncols\do{\advance\@multicnt\@ne
1025 \toks@\expandafter{\the\toks@ &}}%
1026 \toks@\expandafter{\the\toks@ \cr
1027 \noalign{\global\let\@arstrutbox \tabu@arstrutbox}}%
1028 %% \tabu@debug{LIGNE AJOUTEE:\the\toks@}%
1029 \fi\fi
1030 \ifnum0='{ \fi\expandafter}\the\toks@
1031 }% \tabu@ccline
1032 \def\tabu@add #1\@nil{\toks@\expandafter{\the\toks@
1033 \begingroup #1\endgroup\ifmmode$\fi}}

```



<code>\tabu@startstop</code>	<p>This macro parses the mandatory argument of <code>\tabucline</code>: start-column and end-column of the cline.</p> <pre> 1034 \def\tabu@startstop#1{\tabu@start@stop #1\relax 1-\tabu@nbcolls\@nnil} 1035 \def\tabu@start@stop #1-#2\@nnil{% 1036   \@defaultunits \tabu@start \number0#1\relax\@nnil 1037   \@defaultunits \tabu@stop \number0#2\relax\@nnil 1038   \ifnum\tabu@start&gt;\tabu@nbcolls\tabu@start \tabu@nbcolls\else 1039     \ifnum\tabu@start&lt;\@ne \tabu@start \@ne \fi\fi 1040   \ifnum\tabu@stop&gt;\tabu@nbcolls \tabu@stop \tabu@nbcolls \else 1041     \ifnum\tabu@stop&lt;\@ne \tabu@stop \tabu@nbcolls\fi\fi 1042 }% \tabu@start@stop </pre>
<code>\tabu@getlinespec</code>	<p>This macro parses the optional argument of <code>\tabucline</code> and check if it's a line specification (then <code>\tabu@getline</code> is expanded) or a <code>\leaders</code> specification (then <code>\tabu@leaders</code> is expanded).</p> <pre> 1043 \def\tabu@getlinespec#1{\let\tabu@xleaders \relax \let\tabu@color \@empty 1044   \@defaultunits \let\@tempa=#1 \relax\@nnil 1045   \ifx\@tempa\relax \let\tabu@xleaders \tabu@defaultleaders\else 1046   \ifx\@tempa\hbox \tabu@defleaders{#1}\else 1047   \ifx\@tempa\box \tabu@defleaders{#1}\else 1048   \ifx\@tempa\copy \tabu@defleaders{#1}\else 1049   \ifcsname tabu@line@style@\detokenize{#1}\endcsname 1050     \csname tabu@line@style@\detokenize{#1}\endcsname 1051   \else \tabu@getline{#1\p@ on0pt off0pt}\fi\fi\fi\fi\fi 1052 }% \tabu@getlinespec 1053 \def\tabu@defleaders#1{% 1054   \def\tabu@xleaders{\iftabu@colortbl\CT@arc@\fi 1055     \xleaders\hbox{\lower.5\extrarowheight#1}\tabu@leaderfill}} 1056 \def\tabucline@warn#1{\PackageWarning{tabu} 1057   {Undefined line style: #1 1058   \MessageBreak Using default line style instead}}% 1059 \let\tabu@xleaders \tabu@defaultleaders 1060 }% \tabucline@warn </pre>
<code>\tabu@getline</code>	<p>This macro parses the optional argument of <code>\tabucline</code> (or the one of <code>\tabulinestyle</code>) and extract the thickness, the dash and gap specified. Default values assignments are done either.</p>
<code>\tabu@maybecolor</code>	<pre> 1061 \def\tabu@getline#1{\tabu@lineon #1 \@nil on\tabu@linedash \p@ \@nil\@nnil{#1}} 1062 \def\tabu@maybecolor{\ifx\tabu@color\@empty \afterassignment\tabu@colortest 1063   \else \@defaultunits\fi} 1064 \def\tabu@lineon #1on#2\@nil#3\@nnil#4{% 1065   \@defaultunits \let\@tempa=#1 \relax\@nnil 1066   \ifcase 0\ifx o\@tempa 1\else\ifcat A\noexpand\@tempa 2\fi\fi\relax 1067     \tabu@maybecolor \@tempdima #1\arrayrulewidth \p@\@nnil 1068     \tabu@maybecolor \@tempdimb #2\p@ \@nnil 1069     \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil 1070   \or\@tempdima \arrayrulewidth 1071     \tabu@maybecolor \@tempdimb #2\p@ \@nnil 1072     \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil 1073   \else 1074     \tabu@maybecolor \@tempdima \arrayrulewidth #1\p@\@nnil 1075     \ifx\tabu@color\@empty \tabucline@warn\@tempa\else 1076       \@tempdimb \z@ 1077     \tabu@lineoff off0pt \p@\@nnil \fi 1078   \fi 1079 }% \tabu@lineon 1080 \def\tabu@lineoff #1off#2\@nnil{% 1081   \tabu@maybecolor \@tempdimc #2\p@ \@nnil </pre>



```

1082 \iftabu@colortbl \ifx \tabu@color\@empty
1083 \def\tabu@color{\noexpand\CT@arc}%
1084 \fi\fi
1085 \ifdim \@tempdimb=\z@
1086 \ifdim \@tempdimc>\z@ \@tempdimb \tabu@dashgap\relax\fi\fi
1087 \ifdim \@tempdimc=\z@
1088 \ifdim \@tempdimb>\z@ \@tempdimc \tabu@linedash\relax\fi\fi
1089 \ifdim \@tempdima<\z@ \else
1090 \ifdim \@tempdimb<\z@ \else
1091 \ifdim \@tempdimc<\z@ \else
1092 \edef \tabu@xleaders{\tabu@color\xleaders
1093 \ifdim \@tempdimc>\z@
1094 \hbox\bgroup \kern\the\dimexpr \@tempdimc/2\relax\fi
1095 \noexpand\iftabu@firstcline
1096 \vrule depth\dimexpr\the\@tempdima
1097 \ifdim \@tempdimb>\z@ width\the\@tempdimb\fi
1098 \noexpand\else
1099 \vrule height\dimexpr-\extrarowheight+\the\@tempdima
1100 depth\dimexpr\extrarowheight
1101 \ifdim \@tempdimb>\z@ width\the\@tempdimb\fi
1102 \noexpand\fi
1103 \ifdim \@tempdimc>\z@
1104 \kern\the\dimexpr \@tempdimc/2\egroup\fi
1105 \tabu@leaderfill}%
1106 \fi\fi\fi
1107 }% \tabu@lineoff

```

## \tabu@colortest

```

1108 \def\tabu@colortest{\futurelet\tabu@temp\tabu@linecolor}
1109 \def\tabu@linecolor{%
1110 \ifcase 0\if ,\noexpand\tabu@temp\else
1111 \ifx\relax\tabu@temp\else
1112 \ifx \@sptoken\tabu@temp1\else
1113 \ifcat A\noexpand\tabu@temp2\else
1114 3\fi\fi\fi\fi\relax
1115 \def\tabu@next##1{\futurelet\tabu@temp\tabu@linecolor}%
1116 \or\def\tabu@next{\tabu@gobblespace{\futurelet\tabu@temp\tabu@linecolor}}%
1117 \or\let\tabu@next \tabu@getlinecolor
1118 \else\expandafter\remove@to@nnil
1119 \fi \tabu@next
1120 }% \tabu@linecolor
1121 \def\tabu@getlinecolor#1\p@{%
1122 \edef\@tempa{\zap@space #1 \@empty}%
1123 \ifcsname\string\color@\@tempa\endcsname
1124 \edef\tabu@color{\noexpand\noexpand\noexpand\color{\@tempa}}% \set@color
1125 \fi\remove@to@nnil
1126 }% \tabu@getlinecolor

```

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

```

1127 \def\tabulinestyle#1{\@for\@tempa:=#1\do
1128 {\expandafter\tabu@linestyle\@tempa==\@nil}}
1129 \def\tabu@linestyle#1=#2=#3\@nil{%
1130 \begingroup \tabu@getlinespec {#2}%
1131 \toks@\expandafter{\tabu@xleaders}%
1132 \@temptokena\expandafter{\tabu@color}%
1133 \expandafter\xdef\csname tabu@line@style@\detokenize{#1}\endcsname
1134 {\@tempdima\the\@tempdima
1135 \def\noexpand\tabu@color{\the\@temptokena}\relax

```

```

1136      \def\noexpand\tabu@xleaders{\the\toks@}\relax}%
1137    \endgroup
1138 }% \tabu@linestyle
1139 \expandafter\def \csname tabu@line@style\endcsname {%
1140      \let\tabu@xleaders \tabu@defaultleaders}%
1141 \def\tabu@defaultleaders{\iftabu@colortbl\CT@arc@\fi
1142   \leaders
1143   \iftabu@firstcline
1144     \vrule depth \arrayrulewidth
1145   \else
1146     \vrule height\dimexpr-\extrarowheight+\arrayrulewidth
1147           depth \extrarowheight
1148   \fi
1149   \tabu@leaderfill}
1150 \let\tabu@leaderfill \hfil

```

## 10.12 Verbatim inside tabu with X columns

`\tabu@sanitizetext`

```

1151 {\catcode32=13\relax\catcode'\^@=13\relax
1152 \gdef\tabu@verb{\@sanitize\catcode'\^=7\edef\string~\tabu@makeatletter
1153 \catcode32=13\let =\ \catcode'\^@=13\def~@{\hskip\parfillskip\null\linebreak}%
1154 \endlinechar\m@ne}}
1155 {\catcode64=13\edef @{\string @}\expandafter\xdef
1156   \csname tabu@makeatletter\endcsname{\catcode64=13\def\noexpand @{\unskip @}}
1157 \newcommand\tabu@sanitizetext[1][\ttfamily]{\begingroup
1158   \tabu@verb #1\tabu@s@nitizetext}
1159 \long\def\tabu@s@nitizetext#1{\@makeother\{\@makeother\}%
1160   \everyeof{\noexpand}\scantokens{#1}\endgroup}

```

## 10.13 Numbers in tabu

`\tabudecimal`

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

```

1161 \def\tabu@tabudecimal#1{%
1162   \def\tabu@decimal{#1}\@temptokena{}%
1163   \let\tabu@getdecimal@ \tabu@getdecimal@ignorespaces
1164   \tabu@scandecimal
1165 }% \tabu@tabudecimal
1166 \def\tabu@scandecimal{\futurelet \tabu@temp \tabu@getdecimal@}
1167 \def\tabu@skipdecimal#1{#1\tabu@scandecimal}
1168 \def\tabu@getdecimal@ignorespaces{%
1169   \ifcase 0\ifx\tabu@temp\ignorespaces\else
1170     \ifx\tabu@temp\@sptoken1\else
1171       2\fi\fi\relax
1172   \let\tabu@getdecimal@ \tabu@getdecimal
1173   \expandafter\tabu@skipdecimal
1174 \or \expandafter\tabu@gobblespace\expandafter\tabu@scandecimal
1175 \else \expandafter\tabu@skipdecimal
1176 \fi
1177 }% \tabu@getdecimal@ignorespaces
1178 \def\tabu@get@decimal#1{\@temptokena\expandafter{\the\@temptokena #1}%
1179   \tabu@scandecimal}
1180 \def\do#1{%
1181   \def\tabu@get@decimalspace#1{%
1182     \@temptokena\expandafter{\the\@temptokena #1}\tabu@scandecimal}%
1183 }\do{ }

```

```

1184 \let\tabu@@tabudecimal \tabu@tabudecimal

\tabu@getdecimal

1185 \def\tabu@getdecimal{%
1186   \ifcase 0\ifx 0\tabu@temp\else
1187     \ifx 1\tabu@temp\else
1188     \ifx 2\tabu@temp\else
1189     \ifx 3\tabu@temp\else
1190     \ifx 4\tabu@temp\else
1191     \ifx 5\tabu@temp\else
1192     \ifx 6\tabu@temp\else
1193     \ifx 7\tabu@temp\else
1194     \ifx 8\tabu@temp\else
1195     \ifx 9\tabu@temp\else
1196     \ifx .\tabu@temp\else
1197     \ifx ,\tabu@temp\else
1198     \ifx -\tabu@temp\else
1199     \ifx +\tabu@temp\else
1200     \ifx e\tabu@temp\else
1201     \ifx E\tabu@temp\else
1202     \ifx\tabu@celleft\tabu@temp1\else
1203     \ifx\ignorespaces\tabu@temp1\else
1204     \ifx\@sptoken\tabu@temp2\else
1205       3\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\relax
1206     \expandafter\tabu@get@decimal
1207   \or \expandafter\tabu@skipdecimal
1208   \or \expandafter\tabu@get@decimalspace
1209   \else\expandafter\tabu@printdecimal
1210   \fi
1211 }% \tabu@getdecimal
1212 \def\tabu@printdecimal{%
1213   \edef\tabu@temp{\the\@temptokena}%
1214   \ifx\tabu@temp\@empty\else
1215   \ifx\tabu@temp\space\else
1216     \expandafter\tabu@decimal\expandafter{\the\@temptokena}%
1217   \fi\fi
1218 }% \tabu@printdecimal

```

## 10.14 \extrarowsep and \tabulinesep

**\extrarowsep** \extrarowsep makes the assignment for both \extrarowheight and \extrarowdepth.

The macro may be prefixed by \global.

```

1219 \def\extrarowsep{\edef\tabu@C@rowsep{\the\numexpr\tabu@C@rowsep+1}%
1220   \aftergroup\tabu@G@rowsep
1221   \afterassignment\tabu@setrowsep \extrarowheight
1222 }% \extrarowsep
1223 \def\tabu@setrowsep{%
1224   \extrarowdepth\extrarowheight
1225   \tabu@Gsave \tabu@G@rowsep \tabu@C@rowsep \extrarowheight \extrarowdepth
1226 }% \tabu@setrowsep
1227 \def\tabu@G@rowsep{\ifx \tabu@G@rowsep\@empty \else \tabu@R@rowsep\fi}
1228 \def\tabu@R@rowsep{\tabu@G@restore \tabu@G@rowsep \tabu@C@rowsep}
1229 \let\tabu@C@rowsep \z@
1230 \let\tabu@G@rowsep \@empty

```

**\tabulinesep** \tabulinesep makes the assignment for both \abovetabulinesep and \belowtabulinesep, similar to cellspace \cellspacetoplimit and \cellspacebottomlimit.

The macro may be prefixed by `\global`.

```

1231 \def\tabulinesep{\edef\tabu@C@linesep{\the\numexpr\tabu@C@linesep+1}%
1232   \aftergroup\tabu@Glinesep
1233   \afterassignment\tabu@setlinesep \abovetabulinesep
1234 }% \tabulinesep
1235 \def\tabu@setlinesep{%
1236   \belowtabulinesep\abovetabulinesep
1237   \tabu@Gsave \tabu@G@linesep \tabu@C@linesep \abovetabulinesep \belowtabulinesep
1238 }% \tabu@setlinesep
1239 \def\tabu@Glinesep{\ifx \tabu@G@linesep\@empty \else \tabu@Rlinesep\fi}
1240 \def\tabu@Rlinesep{\tabu@Grestore \tabu@G@linesep \tabu@C@linesep}
1241 \let\tabu@C@linesep \z@
1242 \let\tabu@G@linesep \@empty

```

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

```

1243 \def\tabu@Gsave #1#2#3#4{\xdef#1{#1%
1244   \toks#2{\toks\the\currentgrouplevel{\global#3\the#3\global#4\the#4}}}%
1245 }% \tabu@Gsave
1246 \def\tabu@Grestore#1#2{\begingroup
1247   \toks#2{#1\toks\currentgrouplevel\expandafter{\expandafter}\the\toks#2\relax
1248   \ifcat$\the\toks\currentgrouplevel$\else
1249     \global\let#1\@empty \global\let#2\z@
1250   \fi
1251   \expandafter\endgroup
1252   \the\toks\currentgrouplevel
1253 }% \tabu@Grestore

```

`\tabu@verticalmeasure`

```

1254 \def\tabu@verticalmeasure{%
1255   \ifnum \currentgrouptype>12
1256     \begingroup\lrbox\tabu@box \d@llarbegin
1257     \let\tabu@verticalspacing \tabu@verticalsp@lcr
1258   \else
1259     \setbox\tabu@box
1260     \ifnum \currentgrouptype=5\vtop\else\vbox\fi \bgroup
1261     \everypar{\let\@finalstrut \@gobble
1262     \let\tabu@verticalspacing \tabu@verticalsp@pmb
1263   \fi
1264 }% \tabu@verticalmeasure

```

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

```

1265 \def\tabu@verticalsp@lcr{% adjust \ht \tabu@box and \dp \tabu@box
1266   \d@llarend\endlrbox\endgroup
1267   \@tempdima \dimexpr \ht\@arstrutbox -\ht\tabu@box \relax
1268   \@tempdimb \dimexpr \dp\@arstrutbox -\dp\tabu@box \relax
1269   \ifdim \@tempdima<\abovetabulinesep
1270     \ht\tabu@box=\dimexpr \ht\tabu@box+\abovetabulinesep\relax \fi
1271   \ifdim \@tempdimb<\belowtabulinesep
1272     \dp\tabu@box=\dimexpr \dp\tabu@box+\belowtabulinesep \relax \fi
1273   \leavevmode \box\tabu@box
1274 }% \tabu@verticalsp@lcr

```

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

```

1275 \def\tabu@verticalsp@pmb{% inserts struts as needed
1276   \par \expandafter\egroup
1277   \expandafter\@tempdimc \the\prevdepth
1278   \@tempdima \dimexpr \ht\@arstrutbox -\ht\tabu@box \relax

```

```

1279 \@tempdimb \dimexpr \dp\@arstrutbox -\@tempdimc \relax
1280 \ifdim \@tempdima<\abovetabulinesep
1281     \llap {\vrule
1282         \@height \dimexpr\ht\tabu@box+\abovetabulinesep\relax
1283         \@depth -\ht\tabu@box
1284         \@width \tabustrutrule
1285     }\nointerlineskip \vskip-\ht\tabu@box
1286 \fi
1287 \unvbox\tabu@box
1288 \ifdim \@tempdimb<\belowtabulinesep
1289     \llap{\vrule
1290         \@depth \dimexpr \belowtabulinesep \relax
1291         \@height \z@
1292         \@width \tabustrutrule
1293     }\nointerlineskip%\vskip-\@tempdimc
1294     \let\@finalstrut \gobble
1295 \fi
1296 }% \tabu@verticalsp@pmb

```

## 10.15 \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>`.

```

1297 \newcommand*\savetabu[1]{\noalign{%
1298     \ifx\\#1\\\tabu@savewarn{}\{The tabu will not be saved\}\else
1299         \@ifundefined{tabu@saved@\string#1}\{\tabu@savewarn{#1}\{Overwritting\}}%
1300         {%
1301             \toks0\expandafter{\tabu@savedpreamble}%
1302             \toks1\expandafter{\tabu@savedparam}%
1303             \toks2\expandafter{\tabu@savedpream}%
1304             \toks3{}%
1305             \iftabu@negcoef
1306                 \edef\tabu@savewd{\tabu@savewd{\number\numexpr1}}%
1307                 \toks3\expandafter{\tabu@savewd}\fi
1308             \expandafter\xdef\csname tabu@saved@\string#1\endcsname##1{%
1309                 \noexpand\ifcase##1\relax
1310                     \noexpand\ifdim\tabu@target>\z@
1311                         \noexpand\tabu@warn@usetabu
1312                     \noexpand\fi
1313                     \tabu@target \the\tabu@target\relax
1314                     \tabucolX \the\tabucolX\relax
1315                     \the\toks1%
1316                     \gdef\noexpand\tabu@usetabu{%
1317                         \the\toks1%
1318                         \def\noexpand\@halignto{\tabu@halignto}%
1319                         \tabucolX \the\tabucolX\relax
1320                         \def\noexpand\@preamble{\the\toks0}%
1321                         \noexpand\tabu@make@arstrutbox
1322                         \the\toks3
1323                         \noexpand\ifx \noexpand\tabu@align\noexpand\tabu@aligndefault@text
1324                             \def\noexpand\tabu@align{\tabu@align}%
1325                         \noexpand\fi}%
1326                     \aftergroup\noexpand\tabu@usetabu
1327                 \noexpand\else
1328                     \noexpand\ifdim\tabu@target=\z@
1329                         \tabu@target \the\tabu@target\relax
1330                     \noexpand\fi
1331                     \gdef\noexpand\tabu@preamble{%

```

```

1332          \noexpand\ifx \noexpand\tabu@align\noexpand\tabu@aligndefault@text
1333          \def\noexpand\tabu@align{\tabu@align}%
1334          \noexpand\fi}%
1335          \aftergroup\noexpand\tabu@preamble
1336          \noexpand\fi
1337          \the\toks2}%
1338          \tabu@message@save{#1}}%
1339          \fi}%
1340 }% \savetabu
1341 \def\tabu@aligndefault@text{\tabu@aligndefault}%
1342 \def\tabu@warn@usetabu{\PackageWarning{tabu}
1343   {Specifying a target with \string\usetabu\space is useless
1344   \MessageBreak The target cannot be changed!}}
1345 \def\tabu@savewd#1{%
1346   \ifnum#1>\tabu@nbcolls \expandafter \@gobble
1347   \else \ifdim\tabu@wd{#1}>\z@
1348     \noexpand\tabu@wddef{#1}{\tabu@wd{#1}}\fi
1349     \expandafter\tabu@savewd % loop
1350   \fi{#1+1}%
1351 }% \tabu@savewd
1352 \def\tabu@make@arstrutbox{%
1353   \@tempdima \ht \strutbox
1354   \@tempdimb \dp \strutbox
1355   \advance \@tempdima by\extrarowheight
1356   \advance \@tempdimb by\extrarowdepth
1357   \setbox \@arstrutbox \hbox{\vrule
1358     \@height \arraystretch \@tempdima
1359     \@depth \arraystretch \@tempdimb
1360     \@width \z@}%
1361 }% \tabu@make@arstrutbox

```

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

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

```

1362 \def\tabu@savewarn#1#2{\PackageInfo{tabu}
1363   {User-name ‘#1’ already used for \string\savetabu
1364   \MessageBreak #2}}%
1365 \def\tabu@saveerr#1{\PackageError{tabu}
1366   {User-name ‘#1’ is unknown for \string\usetabu
1367   \MessageBreak I cannot restore an unknown preamble!}\@ehd}

```

## 10.16 `\rowfont`

### Setting font and alignment specification

`\rowfont` `\rowfont` uses the control sequences `\tabu@celllalign`, `\tabu@celleft`, `\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@celleft` 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`.

```

1368 \def\tabu@rowfont{\noalign{\ifnum0=‘}\fi\tabu@row@font}
1369 \newcommand*\tabu@row@font[2][{}]{%
1370   \global\let\tabu@rowfontreset \tabu@rowfont@reset

```

```

1371 \global\let\tabu@cellleft \tabu@cellleft
1372 \global\let\tabu@cellright \tabu@cellright
1373 \global\let\tabu@celllalign \tabu@celllalign
1374 \global\let\tabu@cellralign \tabu@cellralign
1375 \gdef\tabu@cellfont{#2}%
1376 \ifcsname tabu@cell@#1\endcsname % row alignment
1377 \csname tabu@cell@#1\endcsname \fi
1378 \toks@\expandafter {\tabu@cellleft\tabu@cellfont}% inside \noalign group ok
1379 \xdef\tabu@cellleft {\the\toks@}%
1380 \ifnum0='{ \fi}% end of noalign group
1381 }% \rowfont
1382 \def\tabu@ifcolorleavevmode#1{\let\color \tabu@leavevmodecolor #1\let\color\tabu@color}%
1383 \AtBeginDocument{%
1384 \ifdefined\color
1385 \let\tabu@color \color
1386 {\toks@\expandafter\expandafter\expandafter{\csname color \endcsname}
1387 \xdef\tabu@leavevmodecolor{\leavevmode\the\toks@}}
1388 \else \let\tabu@ifcolorleavevmode \@firstofone
1389 \fi
1390 }%

```

`\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.

```

1391 \def\tabu@rowfont@reset{%
1392 \noalign{%
1393 \global\let\tabu@rowfontreset \@empty
1394 \global\let\tabu@cellleft \tabu@cellleft
1395 \global\let\tabu@cellright \tabu@cellright
1396 \global\let\tabu@cellfont \@empty
1397 \global\let\tabu@celllalign \tabu@celllalign
1398 \global\let\tabu@cellralign \tabu@cellralign
1399 }%
1400 }% \tabu@rowfont@reset
1401 \let\tabu@rowfontreset \@empty

```

`\everyrow`

```

1402 \def\everyrow #1{%
1403 \ifdim\baselineskip=\z@\noalign\fi
1404 {\gdef\tabu@everyrow@hook{#1}}%
1405 }% \everyrow

```

## 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=\count@=\@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@celleft` to this token (`\toks<i>`). This token is finished and will not change afterwards.

**or:** `i=\count@=\@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 `\count@=\@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 `|\count@-\@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 \@one` and initialize the token to prepare (`\toks\count@=\toks<i+1>`) to an empty one.

```

1406 \newif\iftabu@cellright
1407 \AtBeginDocument{\let\tabu@prepnext@tokORI \prepnext@tok}% original definition
1408 \def\tabu@prepnext@tok{%
1409     \ifnum \count@<\z@ % <first initialisation>
1410         \@tempcnta \@M % <not initialized by array.sty>
1411         \tabu@nbc\z@
1412         \let\tabu@fornoopORI \@fornoop
1413         \expandafter\gdef\expandafter\tabu@global@temp\expandafter{%
1414             \tabu@global@temp\gdef\tabu@global@temp{}\tabu@nbc\z@}%
1415         \expandafter\let\csname tabu@tok@1L\endcsname \relax
1416         \tabu@cellrightfalse
1417     \else
1418         \ifcase \numexpr \count@-\@tempcnta \relax % (case 0): prev. token is left
1419             \advance \tabu@nbc\@ne
1420             \expandafter\let\csname tabu@tok@\the\tabu@nbc R\endcsname \relax
1421             \expandafter\gdef\expandafter\tabu@global@temp\expandafter{%
1422                 \tabu@global@temp \advance\tabu@nbc\@ne}%
1423             \iftabu@cellright % before-previous token is right and is finished
1424                 \tabu@cellrightfalse % <only once>
1425                 \tabu@savetok R\tabu@preptokenright
1426             \fi
1427             \ifnum \tabu@nbc=\@ne \tabu@savetok L\fi
1428             \tabu@preptokenleft
1429         \or % (case 1) previous token is right
1430             \tabu@savetok R\tabu@cellrighttrue
1431             \let\@fornoop \tabu@lastnoop
1432         \else % special column: do not change the token
1433             \ifnum \tabu@nbc>\z@ %special column: always on the right of normal one
1434                 \tabu@savetok R%
1435             \else % unless this is the very first column (\tabu@nbc=0)
1436                 \advance\tabu@nbc\@ne \tabu@savetok L\advance\tabu@nbc\m@ne
1437             \fi
1438             \iftabu@cellright % before-previous token is right
1439                 \tabu@cellrightfalse

```

```

1440          \tabu@preptokenright
1441          \fi
1442          \fi % \ifcase
1443          \fi
1444          \tabu@prepnext@tokORI
1445 }% \tabu@prepnext@tok
1446 \long\def\tabu@lastnoop#1\@#2#3{\in@\@nextchar{#2}%
1447   \ifin@
1448     \let\@fornoop \tabu@fornoopORI
1449     \toks0\expandafter{\expandafter\tabu@everyrowtrue \the\toks0}%
1450     \expandafter\prepnext@tok
1451   \fi
1452 }% \tabu@lastnoop
1453 \def\tabu@preptokenright{%
1454   \advance \count@ \m@ne
1455   \toks\count@\expandafter {\the\toks\count@ \tabu@cellright \tabu@celllralign}%
1456   \advance \count@ \@ne
1457 }% \tabu@preptokenright
1458 \def\tabu@preptokenleft{\toks\count@\expandafter{\expandafter\tabu@celllalign
1459   \the\toks\count@ \tabu@cellleft}% after because of $
1460 }% \tabu@preptokenleft
1461 \def\tabu@savetok#1{\begingroup
1462   \expandafter\tabu@savetok\csname tabu@tok@\the\tabu@nbcols #1\endcsname
1463 }% \tabu@savetok
1464 \def\tabu@savetok#1{%
1465   \ifnum \@chclass=\@ne \toks@\toks\count@
1466   \else\ifnum \@chclass=5 \toks@\toks\count@
1467   \else \toks@{}\fi\fi
1468   \unless\ifx#1\relax
1469     \toks@\expandafter\expandafter\expandafter{%
1470       \expandafter#1\the\toks@}\fi
1471   \toks@\expandafter{\expandafter\def\expandafter#1\expandafter{%
1472     \the\toks@}}%
1473   \@temptokena\expandafter{\tabu@global@temp}%
1474   \xdef\tabu@global@temp {\the\@temptokena \the\toks@}%
1475   \expandafter\endgroup \the\toks@
1476 }% \tabu@savetok

```

## Neutralisation of glues and alignment modification

```

\tabu@cellleft First initialisation to \@empty.
\tabu@celllalign 1477 \let\tabu@cellleft \@empty
\tabu@cellright 1478 \let\tabu@cellright \@empty
\tabu@celllralign 1479 \def\tabu@celllalign {\tabu@cellleft}% row font spec. applies to pre-column material
\tabu@cellfont 1480 \let\tabu@celllralign \@empty
1481 \let\tabu@cellfont \@empty

\tabu@cell@align
1482 \def\tabu@cell@align #1#2#3{%
1483   \toks@\expandafter{\tabu@celllalign #1}%
1484   \xdef\tabu@celllalign{\the\toks@}%
1485   \toks@\expandafter{\tabu@celllralign #2}%
1486   \xdef\tabu@celllralign{\the\toks@}%
1487   \toks@\expandafter{\tabu@cellleft #3}%
1488   \xdef\tabu@cellleft{\the\toks@}%
1489 }% \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
1490 \def\tabu@cell@l{% force alignment to left
1491     \tabu@cell@align
1492     {\tabu@removehfil \raggedright \tabu@cellleft}% left
1493     {\tabu@flush1\tabu@ignorehfil}% right
1494     \raggedright
1495 }% \tabu@cell@l
1496 \def\tabu@cell@c{% force alignment to center
1497     \tabu@cell@align
1498     {\tabu@removehfil \centering \tabu@flush{.5}\tabu@cellleft}
1499     {\tabu@flush{.5}\tabu@ignorehfil}
1500     \centering
1501 }% \tabu@cell@c
1502 \def\tabu@cell@r{% force alignment to right
1503     \tabu@cell@align
1504     {\tabu@removehfil \raggedleft \tabu@flush1\tabu@cellleft}
1505     \tabu@ignorehfil
1506     \raggedleft
1507 }% \tabu@cell@r
1508 \def\tabu@cell@j{% force justification (for p, m, b columns)
1509     \tabu@cell@align
1510     {\tabu@justify\tabu@cellleft}
1511     {}
1512     \tabu@justify
1513 }% \tabu@cell@j
1514 \def\tabu@justify{%
1515     \leftskip\z@skip \@rightskip\leftskip \rightskip\@rightskip
1516     \parfillskip\@flushglue
1517 }% \tabu@justify
1518 %% ragged2e settings
1519 \def\tabu@cell@L{% force alignment to left (ragged2e)
1520     \tabu@cell@align
1521     {\tabu@removehfil \RaggedRight \arraybackslash
1522     \tabu@cellleft}
1523     {\tabu@flush1\tabu@ignorehfil}
1524     {\RaggedRight \arraybackslash}%
1525 }% \tabu@cell@L
1526 \def\tabu@cell@C{% force alignment to center (ragged2e)
1527     \tabu@cell@align
1528     {\tabu@removehfil \Centering \arraybackslash
1529     \tabu@flush{.5}\tabu@cellleft}
1530     {\tabu@flush{.5}\tabu@ignorehfil}
1531     {\Centering \arraybackslash}%
1532 }% \tabu@cell@C
1533 \def\tabu@cell@R{% force alignment to right (ragged2e)
1534     \tabu@cell@align
1535     {\tabu@removehfil \RaggedLeft \arraybackslash
1536     \tabu@flush1\tabu@cellleft}
1537     {\tabu@ignorehfil}
1538     {\RaggedLeft \arraybackslash}%
1539 }% \tabu@cell@R
1540 \def\tabu@cell@J{% force justification (ragged2e)
1541     \tabu@cell@align
1542     {\justifying \arraybackslash \tabu@cellleft}
1543     {}

```

```

1544     {\justifying \arraybackslash}%
1545 }% \tabu@cell@J
1546 \def\tabu@flush#1{%
1547     \iftabu@colortbl      % colortbl uses \hfill rather than \hfil
1548     \hskip \ifnum\currentgrouptype>13 \stretch{#1}%
1549     \else\ifdim#1pt<1pt \tabu@cellskip
1550     \else \stretch{#1}
1551     \fi\fi \relax
1552 \else                    % array.sty
1553     \ifnum \currentgrouptype>13\relax
1554         \hfil \hskip1sp
1555     \fi
1556 \fi
1557 }% \tabu@flush
1558 \AtBeginDocument{%
1559     \ifpackageloaded{ragged2e}
1560     {}
1561     {\let\tabu@cell@L \tabu@cell@l
1562     \let\tabu@cell@R \tabu@cell@r
1563     \let\tabu@cell@C \tabu@cell@c
1564     \let\tabu@cell@J \tabu@cell@j
1565     }%
1566 }% \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.

```

1567 \newskip\tabu@cellskip
1568 \let\tabu@hfil\hfil
1569 \let\tabu@hfill\hfill
1570 \let\tabu@hskip\hskip
1571 \def\tabu@removehfil{%
1572     \iftabu@colortbl
1573     \unkern \tabu@cellskip = \lastskip
1574     \ifnum\gluestretchorder\tabu@cellskip = \tw@ \hskip-\tabu@cellskip
1575     \else \tabu@cellskip = \z@skip
1576     \fi
1577 \else
1578     \ifdim\lastskip=1sp\unskip\fi
1579     \ifnum\gluestretchorder\lastskip = \@ne
1580         \hfilneg % \hfilneg for array.sty but not for colortbl...
1581     \fi
1582 \fi
1583 }% \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.

```

1584 \def\tabu@ignorehfil{%
1585     \aftergroup\tabu@nohfil
1586 }% \tabu@ignorehfil
1587 \def\tabu@nohfil{% \hfil -> do nothing + restore original \hfil
1588     \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1589 }% \tabu@nohfil
1590 \AtBeginDocument{%
1591     \ifpackageloaded{colortbl}
1592     {%
1593     \def\tabu@nohfil{%
1594         \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group

```

```

1595      \def\hfill{\let\hfill\tabu@hfill}% (colortbl uses \hfill) pfff...
1596      \def\hskip##1\relax{\let\hskip\tabu@hskip}}% local
1597  }% @ifpackageloaded colortbl
1598  {}%
1599 }% AtBeginDocument

```

## 10.17 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.

```

1600 \def\tabu@fbox{%
1601   \leavevmode
1602   \let\color@bgroup\bgroup
1603   \def\color@egroup{\endgraf\egroup}%
1604   \afterassignment\tabu@begin@fbox
1605           \setbox\@tempboxa \hbox
1606 }% \tabu@fbox
1607 \def\tabu@begin@fbox{\color@bgroup\kern\fboxsep\aftergroup\tabu@end@fbox}
1608 \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.

```

1609 \def\tabu@color@b@x#1#2{\leavevmode
1610   \def\tabu@do@color@b@x{#1#2\color@block{\wd\z@}{\ht\z@}{\dp\z@}\box\z@}}%
1611   \afterassignment\tabu@begin@color@b@x
1612           \setbox\z@ \hbox
1613 }% \tabu@color@b@x
1614 \def\tabu@begin@color@b@x{%
1615   \bgroup \kern\fboxsep \set@color
1616   \aftergroup\tabu@end@color@b@x
1617 }% \tabu@begin@color@b@x
1618 \def\tabu@end@color@b@x{\kern\fboxsep \egroup
1619   \dimen@ \ht\z@ \advance\dimen@\fboxsep \ht\z@ \dimen@
1620   \dimen@ \dp\z@ \advance\dimen@\fboxsep \dp\z@ \dimen@
1621   \tabu@do@color@b@x
1622 }% \tabu@end@color@b@x

```

### \centering, \raggedright, \raggedleft

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

```

1623 \def\tabu@temp#1#2#3{\toks@\expandafter{#2#3}\xdef#1{\the\toks@}}
1624 \tabu@temp \tabu@centering \centering \arraybackslash
1625 \tabu@temp \tabu@raggedleft \raggedleft \arraybackslash
1626 \tabu@temp \tabu@raggedright \raggedright \arraybackslash
1627 \def\tabu@normalcrbackslash{\let\\\@normalcr}
1628 \def\tabu@trivlist{\expandafter\def\expandafter\@trivlist\expandafter{%
1629   \expandafter\tabu@normalcrbackslash \@trivlist}}%

```

## 10.18 Corrections

### delarray comptability 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.

```

1630 \def\tabu@fix@arrayright{%
1631   \@ifpackageloaded{arydshln}
1632   {%
1633     \@ifpackageloaded{colortbl}
1634     {% colortbl + arydshln
1635       \def\tabu@endarray{%
1636         \adl@endarray \egroup \adl@arrayrestore \CT@end \egroup %<original>
1637         \@arrayright      % <FC>
1638         \gdef\@preamble{}% <FC>
1639         }}% \endarray
1640     {% arydshln / no colortbl
1641       \def\tabu@endarray{%
1642         \adl@endarray \egroup \adl@arrayrestore \egroup %<original>
1643         \@arrayright      % <FC>
1644         \gdef\@preamble{}% <FC>
1645         }}% \endarray
1646   }%
1647   {%
1648     \@ifpackageloaded{colortbl}
1649     {% colortbl / no arydshln
1650       \def\tabu@endarray{%
1651         \crrc \egroup \egroup
1652         \@arrayright % <FC>
1653         \gdef\@preamble{}\CT@end
1654         }}%
1655     {\PackageWarning{tabu}
1656       {\string\@arrayright\space is missing from the
1657       \MessageBreak definition of \string\endarray.
1658       \MessageBreak Comptability with delarray.sty is broken.}}%
1659   }%
1660 }% \tabu@fix@arrayright

```

### arydshln @ columns

```

1661 \def\tabu@adl@xarraydashrule#1#2#3{%
1662   \ifnum\@lastchclass=\adl@class@start\else
1663   \ifnum\@lastchclass=\@ne\else
1664   \ifnum\@lastchclass=5 \else % <FC> @-arg (class 5) and !-arg (class 1)
1665     \adl@leftrulefalse \fi\fi % must be treated the same
1666   \fi
1667   \ifadl@zwvrule\else \ifadl@inactive\else
1668     \addtopreamble{\vrule\@width\arrayrulewidth
1669       \@height\z@ \@depth\z@}\fi \fi
1670   \ifadl@leftrule
1671     \addtopreamble{\adl@vlineL{\CT@arc@}{\adl@dashgapcolor}%
1672       {\number#1}#3}%
1673   \else \addtopreamble{\adl@vlineR{\CT@arc@}{\adl@dashgapcolor}%
1674     {\number#2}#3}
1675   \fi

```

```
1676 }% \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.

```
1677 \def\tabu@adl@act@endpbox{%
1678   \unskip \ifhmode \nobreak \fi   \@finalstrut \@arstrutbox
1679   \egroup \egroup
1680   \adl@colhtdp \box\adl@box \hfil
1681 }% \tabu@adl@act@endpbox
1682 \def\tabu@adl@fix{%
1683   \let\adl@xarraydashrule \tabu@adl@xarraydashrule % <fix> arydshln
1684   \let\adl@act@endpbox    \tabu@adl@act@endpbox    % <fix> arydshln
1685   \let\adl@act@@endpbox   \tabu@adl@act@endpbox    % <fix> arydshln
1686   \let\@preamerror        \@preamerr               % <fix> arydshln
1687 }% \tabu@adl@fix
```

### longtable \@startpbox: \everypar needed

\tabu@LT@startpbox The leading strut should be inserted at \everypar in order for \tabulinesep to work (otherwise, T<sub>E</sub>X is in horizontal mode and \nointerlineskip breaks).

```
1688 \def\tabu@LT@startpbox #1{%
1689   \bgroup
1690   \let\@footnotetext\LT@p@fnttext
1691   \setlength\hsize{#1}%
1692   \@arrayparboxrestore
1693   \everypar{%
1694     \vrule \@height \ht\@arstrutbox \@width \z@
1695     \everypar{}}%
1696 }% \tabu@LT@startpbox
1697 </package>
```



## 11 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 [FC](#)  
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)

## 12 History

[2011/01/19 v2.1]

- [Vertical spacing](#) had a bug with longtabu and paragraph columns. Fixed.
- New: `\everyrow`.
- Fix a bug of `\rowfont` when using [siunitx](#) S columns.
- Some code optimisation.
- To do (if possible): a syntax `X[6mc]S[...]` to “embed” [siunitx](#) S column inside tabu and longtabu X columns...

[2011/01/18 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 pdfT<sub>E</sub>X 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 `\mathscr-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.

## 13 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols	
<code>\@arrayparboxrestore</code> . . . . .	1692
<code>\@arrayright</code> 65, 604, 724, 1637, 1643, 1652, 1656	
<code>\@arstrutbox</code> . . . . .	105, 180, 181, 183, 186, 187, 189, 618, 619, 881, 929, 935, 941, 1002, 1003, 1027, 1267, 1268, 1278, 1279, 1357, 1678, 1694
<code>\@chclass</code> . . . . .	1465, 1466
<code>\@currentHref</code> . . . . .	436
<code>\@currentlabelname</code> . . . . .	437
<code>\@currenvir</code> . . . . .	111, 112
<code>\@defaultunits</code> . . . . .	1036, 1037, 1044, 1063, 1065
<code>\@elt</code> . . . . .	610, 611
<code>\@finalstrut</code> . . . . .	881, 1261, 1294, 1678
<code>\@firstoftwo</code> . . . . .	358, 857
<code>\@flushglue</code> . . . . .	232, 235, 236, 867, 1516
<code>\@footnotemark</code> . . . . .	409
<code>\@footnotetext</code> . . . . .	310, 603, 1690
<code>\@fornoop</code> . . . . .	1412, 1431, 1448
<code>\@gobbletwo</code> . . . . .	256, 835
<code>\@halignto</code> . . . . .	142, 209, 222, 595, 726, 1318
<code>\@ifnextchar</code> . . . . .	19, 108
<code>\@let@token</code> 337, 339, 340, 341, 342, 343, 493, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512	
<code>\@m</code> . . . . .	809
<code>\@makeother</code> . . . . .	1159
<code>\@mkpream</code> . . . . .	247
<code>\@mpfn</code> . . . . .	397, 406, 415
<code>\@multicnt</code> . . . . .	998, 1004, 1008, 1009, 1011, 1016, 1017, 1019, 1020, 1024
<code>\@normalcr</code> . . . . .	313, 1627
<code>\@preamble</code> . . . . .	178, 625, 631, 635, 1320, 1638, 1644, 1653
<code>\@preamerr</code> . . . . .	1686
<code>\@preamerror</code> . . . . .	1686
<code>\@rightskip</code> . . . . .	1515
<code>\@sanitize</code> . . . . .	1152
<code>\@secondoftwo</code> . . . . .	860
<code>\@spaces</code> . . . . .	804
<code>\@sptoken</code> . . . . .	227, 341, 509, 1112, 1170, 1204
<code>\@tempcnta</code> . . . . .	597, 1410, 1418
<code>\@tempdimb</code> . . . . .	1068, 1071, 1076, 1085, 1086, 1088, 1090, 1097, 1101, 1268, 1271, 1279, 1288, 1354, 1356, 1359
<code>\@tempdimc</code> . . . . .	1081, 1086, 1087, 1088, 1091, 1093, 1094, 1103, 1104, 1277, 1279, 1293
<code>\@temptokena</code> 371, 385, 391, 911, 1132, 1135, 1162, 1178, 1182, 1213, 1216, 1473, 1474	
<code>\@thefnmark</code> . . . . .	407, 417
<code>\@trivlist</code> . . . . .	1628, 1629
<code>\@undefined</code> . . . . .	6, 584
<code>\@whilenum</code> . . . . .	937, 1004, 1024
<code>\@whilesw</code> . . . . .	732
<code>\@xfootnote</code> . . . . .	311, 402, 603
<code>\@xfootnotetext</code> . . . . .	310
<code>\{</code> . . . . .	1159
<code>\}</code> . . . . .	1159
<code>^</code> . . . . .	656, 1151, 1152, 1153

<code>\_</code> .....	1153	<code>\delarray_</code> (package_option) .....	48
<b>A</b>		<code>\detokenize</code> ..	833, 965, 968, 1049, 1050, 1133
<code>\abovetabulinesep</code> .....	14, 93, 304, 618, 1233, 1236, 1237, 1269, 1270, 1280, 1282	<code>\dimen@</code> .....	687, 692, 710, 728, 1619, 1620
<code>\adl@act@@endpbox</code> .....	1685	<code>\doublerulesep</code> .....	379, 822
<code>\adl@act@endpbox</code> .....	1684	<code>\dp</code> .....	183, 189, 1265, 1268, 1272, 1279, 1354, 1610, 1620
<code>\adl@arrayrestore</code> .....	1636, 1642	<b>E</b>	
<code>\adl@box</code> .....	1680	<code>\ecolumn</code> .....	874
<code>\adl@class@start</code> .....	1662	<code>\endarray</code> 65, 66, 271, 286, 671, 1639, 1645, 1657	
<code>\adl@colhtdp</code> .....	1680	<code>\endlinechar</code> .....	230, 233, 948, 1154
<code>\adl@dashgapcolor</code> .....	1671, 1673	<code>\endlongtable</code> .....	278
<code>\adl@endarray</code> .....	1636, 1642	<code>\endlongtabu</code> .....	278
<code>\adl@leftrulefalse</code> .....	1665	<code>\endlrbox</code> .....	1266
<code>\adl@ncol</code> .....	682	<code>\endtabu</code> .....	270
<code>\adl@vlineL</code> .....	1671	<code>\endtabular</code> .....	272
<code>\adl@vlineR</code> .....	1673	<code>\everycr</code> .....	318, 910, 911
<code>\adl@xarraydashrule</code> .....	1683	<code>\everyeof</code> .....	1160
<code>\afterassignment</code> .....	20, 21, 260, 354, 546, 911, 1062, 1221, 1233, 1604, 1611	<code>\everypar</code> ....	280, 296, 297, 1261, 1693, 1695
<code>\aftergroup</code> .....	165, 213, 214, 215, 218, 244, 247, 264, 268, 335, 868, 1220, 1232, 1326, 1335, 1585, 1607, 1616	<code>\everyrow</code> .....	12, 600, 1402
<code>\array</code> .....	271	<code>\extracolsep</code> .....	220, 232, 235, 236
<code>\arraybackslash</code> 866, 1521, 1524, 1528, 1531, 1535, 1538, 1542, 1544, 1624, 1625, 1626		<code>\extrarowdepth</code> .....	. 93, 185, 189, 381, 824, 1224, 1225, 1356
<code>\arraycolsep</code> .....	376	<code>\extrarowheight</code> 13, 380, 823, 935, 1055, 1099, 1100, 1146, 1147, 1221, 1224, 1225, 1355	
<code>\arrayrulewidth</code> .....	378, 821, 962, 977, 1067, 1070, 1074, 1144, 1146, 1668	<code>\extrarowsep</code> .....	13, 1219
<code>\arraystretch</code> .....	189, 383, 827, 1358, 1359	<code>\extratabsurround</code> .....	382, 825
<code>\AtEndOfPackage</code> .....	6, 41, 44, 49, 52	<b>F</b>	
<b>B</b>		<code>\footnotetext</code> .....	397, 401
<code>\baselineskip</code> .....	1403	<code>\futurelet</code> .....	225, 337, 355, 493, 970, 1108, 1115, 1116, 1166
<code>\bcolumn</code> .....	874	<b>G</b>	
<code>\begin</code> .....	4, 4	<code>\GenericError</code> .....	266, 602
<code>\belowtabulinesep</code> .....	93, 304, 619, 1236, 1237, 1271, 1272, 1288, 1290	<code>\gluestretchorder</code> .....	1574, 1579
<b>C</b>		<b>H</b>	
<code>\c@footnote</code> .....	403	<code>\H@@footnotetext</code> .....	429
<code>\c@page</code> .....	464	<code>\hbadness</code> .....	597
<code>\catcode</code> 11, 13, 656, 1151, 1152, 1153, 1155, 1156		<code>\hfilneg</code> .....	1580
<code>\cellspacetoplimit</code> .....	869	<code>\hfuzz</code> .....	294, 598
<code>\Centering</code> .....	523, 1528, 1531	<code>\ht</code> 181, 187, 935, 1003, 1265, 1267, 1270, 1278, 1282, 1283, 1285, 1353, 1610, 1619, 1694	
<code>\centering</code> ...	311, 519, 523, 1498, 1500, 1624	<code>\Hy@footnote@currentHref</code> .....	420
<code>\chardef</code> .....	105	<code>\Hy@raisedlink</code> .....	433
<code>\cl@ckpt</code> .....	611	<code>\hyper@@anchor</code> .....	431, 434
<code>\color</code> .....	831, 1124, 1382, 1384, 1385	<b>I</b>	
<code>\color@</code> .....	1123	<code>\ifadl@inactive</code> .....	1667
<code>\color@b@x</code> .....	315	<code>\ifadl@leftrule</code> .....	1670
<code>\color@begingroup</code> .....	869	<code>\ifadl@zwvrule</code> .....	1667
<code>\color@bgroup</code> .....	1602, 1607	<code>\ifcase</code> .....	304, 327, 339, 496, 582, 689, 916, 1011, 1016, 1066, 1110, 1169, 1186, 1309, 1418, 1442
<code>\color@block</code> .....	1610	<code>\ifcat</code> ....	172, 249, 512, 976, 1066, 1113, 1248
<code>\color@egroup</code> .....	1603, 1608	<code>\ifcsname</code> .	443, 732, 851, 965, 1049, 1123, 1376
<code>\copy</code> .....	1048	<code>\ifdim</code> .....	179, 185, 219, 295, 299, 304, 362, 459, 554, 557, 575, 576, 618, 619, 689, 690, 692, 693, 702, 706, 710, 714, 725, 733, 736, 739, 750, 753, 757, 761, 776, 858, 883, 887, 889, 890, 1085, 1086, 1087, 1088, 1089, 1090, 1091, 1093, 1097, 1101, 1103, 1269, 1271, 1280, 1288, 1310, 1328, 1347, 1403, 1549, 1578
<code>\crr</code> .....	682, 928, 994, 1651	<code>\iffalse</code> .....	103, 579
<code>\CT@drsc@</code> .....	373, 830		
<code>\currentgrouplevel</code> ....	1244, 1247, 1248, 1252		
<code>\currentgrouptype</code> ....	1255, 1260, 1548, 1553		
<b>D</b>			
<code>\d@llarbegin</code> .....	1256		
<code>\d@llarend</code> .....	172, 249, 1266		

<code>\ifHy@nesting</code>	430	<code>\newskip</code>	1567
<code>\ifin@</code>	66, 475, 849, 857, 859, 1447	<code>\newtoks</code>	394
<code>\ifnum</code>	26, 31, 579, 617, 630, 694, 713, 723, 791, 807, 808, 809, 811, 812, 935, 994, 999, 1008, 1011, 1016, 1019, 1030, 1038, 1039, 1040, 1041, 1255, 1260, 1346, 1368, 1380, 1409, 1427, 1433, 1465, 1466, 1548, 1553, 1574, 1579, 1662, 1663, 1664	<code>\noalign</code>	682, 915, 924, 928, 940, 994, 1002, 1027, 1297, 1368, 1378, 1392, 1403
<code>\iftabu@cellright</code>	1406, 1423, 1438	<code>\nointerlineskip</code>	1285, 1293
<code>\iftabu@colortbl</code>	<a href="#">259</a> , 371, 384, 828, 930, 943, 961, 963, 988, 1054, 1082, 1141, 1547, 1572	<code>\null</code>	1153
<code>\iftabu@everyrow</code>	101, 102, 103, 923	<code>\numexpr</code>	794, 795, 796, 1219, 1231, 1306, 1418
<code>\iftabu@firstcline</code>	<a href="#">97</a> , 1003, 1095, 1143	<b>O</b>	
<code>\iftabu@measuring</code>	<a href="#">97</a> , 451, 649, 675, 719	<code>\omit</code>	1007
<code>\iftabu@negcoef</code>	<a href="#">97</a> , 213, 582, 688, 843, 1305	<code>\on@line</code>	464, 790, 837
<code>\iftabu@spread</code>	<a href="#">97</a> , 457, 582, 646, 672, 856	<b>P</b>	
<code>\iftrue</code>	102	<code>\PackageError</code>	59, 61, 277, 283, 1365
<code>\ignorespaces</code>	1169, 1203	<code>\PackageInfo</code>	1362
<code>\immediate</code>	262, 264, 268	<code>\PackageWarning</code>	75, 269, 534, 541, 1056, 1342, 1655
<code>\in@</code>	65, 475, 848, 1446	<code>\parfillskip</code>	867, 1153, 1516
<code>\in@false</code>	858	<code>\pdfadjustspacing</code>	875, 876
<b>J</b>		<code>\pdfelapsedtime</code>	28, 29, 794, 795, 796
<code>\justifying</code>	525, 1542, 1544	<code>\preamble</code>	16, 152, 158, 168
<b>L</b>		<code>\preamble_(private_column_type)</code>	<a href="#">152</a>
<code>\lastbox</code>	298	<code>\prepnext@tok</code>	53, 317, 1407, 1450
<code>\lastskip</code>	1573, 1578, 1579	<code>\prevdepth</code>	1277
<code>\leaders</code>	1142	<code>\ProcessOptions</code>	63
<code>\leftskip</code>	1515	<code>\protected</code>	263, 267
<code>\light_(package_option)</code>	<a href="#">51</a>	<code>\protected@xdef</code>	407, 416
<code>\linebreak</code>	1153	<b>Q</b>	
<code>\linegoal</code>	46, 463	<code>\q</code>	967
<code>\linegoal_(package_option)</code>	<a href="#">42</a>	<b>R</b>	
<code>\linewidth</code>	42, 46, 290, 295, 299, 462, 601	<code>\RaggedLeft</code>	524, 1535, 1538
<code>\llap</code>	1281, 1289	<code>\raggedleft</code>	312, 520, 524, 1504, 1506, 1625
<code>\LNGL@setlinegoal</code>	357, 358	<code>\RaggedRight</code>	522, 1521, 1524
<code>\longtable</code>	278	<code>\raggedright</code>	312, 518, 522, 1492, 1494, 1626
<code>\longtabu</code>	71, 112, 276, 282	<code>\remove@to@nnil</code>	514, 1118, 1125
<code>\loop</code>	1006	<code>\repeat</code>	1023
<code>\lower</code>	1055	<code>\rightskip</code>	1515
<code>\lrbox</code>	1256	<code>\romannumeral</code>	967
<code>\LT@bchunk</code>	628, 632, 637	<code>\rowfont</code>	<a href="#">12</a> , 60, 316, 317, <a href="#">1368</a>
<code>\LT@echunk</code>	642	<b>S</b>	
<code>\LT@get@widths</code>	644	<code>\save@decl</code>	53, 173, 250
<code>\LT@output</code>	638	<code>\savetabu</code>	<a href="#">15</a> , 366, 600, 722, 837, 838, <a href="#">1297</a> , 1363
<code>\LT@p@ftntext</code>	1690	<code>\scantokens</code>	230, 233, 948, 1160
<code>\LT@startpbox</code>	280	<code>\set@color</code>	1124, 1615
<code>\LT@chunksz</code>	636	<code>\setbox</code>	180, 186, 298, 618, 619, 622, 634, 643, 863, 1259, 1357, 1605, 1612
<b>M</b>		<code>\setlength</code>	460, 1691
<code>\maxdimen</code>	598	<code>\span</code>	1022
<code>\message</code>	27, 31	<code>\stretch</code>	1548, 1550
<code>\minrowclearance</code>	384, 826, 931, 932, 944	<code>\strip@pt</code>	453, 467, 561
<code>\multicolumn</code>	56, 293	<code>\strutbox</code>	1353, 1354
<code>\multispan</code>	247, 254, 256	<b>T</b>	
<b>N</b>		<code>\tabcolsep</code>	377
<code>\narrowragged</code>	866	<code>\tabu</code>	111, <a href="#">270</a>
<code>\NC@find</code>	143, 174, 201, 205, 208, 232, 235, 236, 251, 387, 833, 956	<code>\tabu@</code>	271, 272, 278, 368
<code>\NC@rewrite@X</code>	163	<code>\tabu@@begin</code>	363, 365, 369
<code>\newcount</code>	81, 82, 83, 84, 85, 86	<code>\tabu@@cellalign</code>	1373, 1397
<code>\newdimen</code>	87, 88, 89, 90, 91, 92, 93, 94, 95, 96	<code>\tabu@@cellleft</code>	1371, 1394
<code>\newsavebox</code>	104	<code>\tabu@@cellralign</code>	1374, 1398
		<code>\tabu@@cellright</code>	1372, 1395
		<code>\tabu@@rewrite@X</code>	474, 864
		<code>\tabu@@tabudecimal</code>	118



<code>\tabu@add</code>	1009, 1017, 1032	<code>\tabu@elapsedtime</code>	30, 791
<code>\tabu@adl@act@endpbox</code>	1677, 1681, 1684, 1685	<code>\tabu@end@color@b@x</code>	1616, 1618, 1622
<code>\tabu@adl@endTRIAL</code>	70, 670, 682	<code>\tabu@end@fbox</code>	1607, 1608
<code>\tabu@adl@fix</code>	70, 286, 1682, 1687	<code>\tabu@endarray</code>	66, 286, 1635, 1641, 1650
<code>\tabu@adl@xarraydashrule</code>	1661, 1676, 1683	<code>\tabu@endenvir</code>	653, 660, 679
<code>\tabu@align</code>	367, 368, 1003, 1323, 1324, 1332, 1333	<code>\tabu@endrewrite</code>	193, 205, 206
<code>\tabu@aligndefault</code>	290, 294, 363, 1341	<code>\tabu@endTRIAL</code>	623, <u>669</u>
<code>\tabu@aligndefault@text</code>	1323, 1332, 1341	<code>\tabu@everycr</code>	318, 910, 912
<code>\tabu@arith</code>	647, 673, <u>683</u>	<code>\tabu@everyrow</code>	912, 914, 926
<code>\tabu@arith@negcoef</code>	688, 730, 745	<code>\tabu@everyrow@hook</code>	323, 325, 924, 927, 1404
<code>\tabu@arrayleft@measure</code>	273, 279, 592	<code>\tabu@everyrowfalse</code>	103, 924
<code>\tabu@arrayleftmeasure</code>	273, <u>615</u>	<code>\tabu@everyrowtrue</code>	102, 326, 908, 1449
<code>\tabu@arstrutbox</code>	105, 941, 1027	<code>\tabu@extracol@sep</code>	225, 226, 228, 239
<code>\tabu@begin</code>	<u>337</u>	<code>\tabu@extracolsep</code>	220, 225
<code>\tabu@begin@color@b@x</code>	1611, 1614, 1617	<code>\tabu@fbox</code>	315, 1600, 1606
<code>\tabu@begin@fbox</code>	1604, 1607	<code>\tabu@first@cline</code>	909, 916, 917, 918, 920, 945
<code>\tabu@box</code>	<u>104</u> , 622, 634, 645, 689, 690, 699, 704, 711, 713, 750, 752, 761, 772, 777, 781, 786, 863, 881, 882, 885, 890, 891, 898, 1256, 1259, 1265, 1267, 1268, 1270, 1272, 1273, 1278, 1282, 1283, 1285, 1287	<code>\tabu@firstcline</code>	55
<code>\tabu@C@linesep</code>	1231, 1237, 1240, 1241	<code>\tabu@firstclinefalse</code>	919
<code>\tabu@C@rowsep</code>	1219, 1225, 1228, 1229	<code>\tabu@firstclinettrue</code>	908
<code>\tabu@cell@align</code>	<u>1482</u> , 1491, 1497, 1503, 1509, 1520, 1527, 1534, 1541	<code>\tabu@fix@arrayright</code>	67, 1630, 1660
<code>\tabu@cell@C</code>	1526, 1532, 1563	<code>\tabu@flush</code>	1493, 1498, 1499, 1504, 1523, 1529, 1530, 1536, 1546, 1557
<code>\tabu@cell@c</code>	<u>1490</u>	<code>\tabu@footnotetext</code>	<u>395</u>
<code>\tabu@cell@J</code>	1540, 1545, 1564	<code>\tabu@footnotes</code>	292, 328, <u>394</u> , 396, 398, 400, 413, 423
<code>\tabu@cell@j</code>	<u>1490</u>	<code>\tabu@footnotetext</code>	310, 395, 445
<code>\tabu@cell@L</code>	1519, 1525, 1561	<code>\tabu@fornoopORI</code>	1412, 1448
<code>\tabu@cell@l</code>	<u>1490</u>	<code>\tabu@G@linesep</code>	1237, 1239, 1240, 1242
<code>\tabu@cell@R</code>	1533, 1539, 1562	<code>\tabu@G@rowsep</code>	1225, 1227, 1228, 1230
<code>\tabu@cell@r</code>	<u>1490</u>	<code>\tabu@GenericError</code>	263, 602
<code>\tabu@cellfont</code>	1375, 1378, 1396, <u>1477</u>	<code>\tabu@GenericErrorORI</code>	265, 266
<code>\tabu@celllalign</code>	305, 306, 1373, 1397, 1458, <u>1477</u> , 1483, 1484	<code>\tabu@get@decimal</code>	1178, 1206
<code>\tabu@celleft</code>	1202, 1371, 1378, 1379, 1394, 1459, <u>1477</u> , 1487, 1488, 1492, 1498, 1504, 1510, 1522, 1529, 1536, 1542	<code>\tabu@get@decimalspace</code>	1181, 1208
<code>\tabu@cellralign</code>	305, 307, 1374, 1398, 1455, <u>1477</u> , 1485, 1486	<code>\tabu@getdecimal</code>	1172, <u>1185</u>
<code>\tabu@cellright</code>	1372, 1395, 1455, <u>1477</u>	<code>\tabu@getdecimal@</code>	1163, 1166, 1172
<code>\tabu@cellrightfalse</code>	1416, 1424, 1439	<code>\tabu@getdecimal@ignorespaces</code>	1163, 1168, 1177
<code>\tabu@cellrighttrue</code>	1430	<code>\tabu@getline</code>	1051, <u>1061</u>
<code>\tabu@cellskip</code>	1549, 1567, 1573, 1574, 1575	<code>\tabu@getlinecolor</code>	978, 1117, 1121, 1126
<code>\tabu@centering</code>	311, 1624	<code>\tabu@getlinespec</code>	998, <u>1043</u> , 1130
<code>\tabu@checklast</code>	194, 196, 204, 206	<code>\tabu@gettarget</code>	352, 353, 354
<code>\tabu@cline</code>	995, 996, 997, 1031	<code>\tabu@Glinesep</code>	1232, 1239
<code>\tabu@cnt</code>	<u>81</u> , 546, 549, 591, 684, 694, 713, 811	<code>\tabu@global@temp</code>	164, 165, 1413, 1414, 1421, 1422, 1473, 1474
<code>\tabu@collect</code>	626, 639, <u>656</u>	<code>\tabu@global@X</code>	210, 243, 244, 287, 452, 466, 594, 611, 621, 632, 636, 720
<code>\tabu@collectbody</code>	656, 657, 664, 666	<code>\tabu@global@X@nested</code>	240, 242
<code>\tabu@color</code>	975, 987, 989, 1043, 1062, 1075, 1082, 1083, 1092, 1124, 1132, 1135, 1382, 1385	<code>\tabu@global@Xnested</code>	212, 240, 245, 617, 630
<code>\tabu@color@b@x</code>	315, 1609, 1613	<code>\tabu@gobble@pt</code>	108, 109
<code>\tabu@colortblfalse</code>	68	<code>\tabu@gobbleopt</code>	<u>107</u> , 478
<code>\tabu@colortbltrue</code>	68	<code>\tabu@gobblespace</code>	<u>107</u> , 228, 346, 1116, 1174
<code>\tabu@colortest</code>	1062, <u>1108</u>	<code>\tabu@Grestore</code>	1228, 1240, <u>1243</u>
<code>\tabu@dashgap</code>	993, 1069, 1072, 1086	<code>\tabu@Growsep</code>	1220, 1227
<code>\tabu@debug</code>	31, 32, 36, 892, 1028	<code>\tabu@Gsave</code>	1225, 1237, <u>1243</u>
<code>\tabu@decimal</code>	1162, 1216	<code>\tabu@halignto</code>	211, 222, 352, 353, 362, 387, 471, 766, 1318
<code>\tabu@defaultleaders</code>	1045, 1059, 1140, 1141	<code>\tabu@header</code>	698, 770, 784, 811
<code>\tabu@defleaders</code>	1046, 1047, 1048, 1053	<code>\tabu@hfil</code>	1568, 1588, 1594
<code>\tabu@dimexpr</code>	161, 484, 571, 574, 577, 848, 857, 860	<code>\tabu@hfill</code>	1569, 1595
<code>\tabu@do@color@b@x</code>	1610, 1621	<code>\tabu@hfuzz</code>	294, 689, 700, 788, 790
		<code>\tabu@hskip</code>	1570, 1596
		<code>\tabu@Hy@footnotetext</code>	418, 428, 441
		<code>\tabu@Hy@ftntext</code>	<u>403</u>
		<code>\tabu@Hy@ftntxt</code>	403, 409, 411, 427
		<code>\tabu@Hy@xfootnote</code>	<u>403</u>
		<code>\tabu@ifcolorleavevmode</code>	1382, 1388

<code>\tabu@ifenvir</code> . . . . .	107, 253	<code>\tabu@nohfil</code> . . . . .	1585, 1587, 1589, 1593
<code>\tabu@ignorehfil</code> . . . . .	1493, 1499, 1505, 1523, 1530, 1537, 1584	<code>\tabu@nolongtabu</code> . . . . .	71, 283
<code>\tabu@immediate</code> . . . . .	262, 265, 269	<code>\tabu@normalcrbackslash</code> . . . . .	1627, 1629
<code>\tabu@islast</code> . . . . .	166, 198, 200	<code>\tabu@norowfont</code> . . . . .	54, 59
<code>\tabu@justify</code> . . . . .	521, 525, 1510, 1512, 1514, 1517	<code>\tabu@nowrite</code> . . . . .	260, 602
<code>\tabu@lastnoop</code> . . . . .	1431, 1446, 1452	<code>\tabu@preamble</code> . . . . .	1331, 1335
<code>\tabu@lastrewrite</code> . . . . .	199, 207, 224	<code>\tabu@prep@TRIAL</code> . . . . .	214, 578
<code>\tabu@lasttry</code> . . . . .	591, 690, 711, 713	<code>\tabu@prepare@save</code> . . . . .	372, 373, 389, 393
<code>\tabu@leaderfill</code> . . . . .	1055, 1105, 1149, 1150	<code>\tabu@preparesave</code> . . . . .	366, 370
<code>\tabu@leavevmodecolor</code> . . . . .	1382, 1387	<code>\tabu@prepnext@tok</code> . . . . .	53, 317, 1406
<code>\tabu@linecolor</code> . . . . .	1108, 1109, 1115, 1116, 1120	<code>\tabu@prepnext@tokORI</code> . . . . .	1407, 1444
<code>\tabu@linedash</code> . . . . .	992, 1061, 1088	<code>\tabu@preptokenleft</code> . . . . .	1428, 1458, 1460
<code>\tabu@linegoal@target</code> . . . . .	355, 356, 360	<code>\tabu@preptokenright</code> . . . . .	1425, 1440, 1453, 1457
<code>\tabu@linegoaltarget</code> . . . . .	354, 355	<code>\tabu@printdecimal</code> . . . . .	1209, 1212, 1218
<code>\tabu@lineoff</code> . . . . .	1069, 1072, 1077, 1080, 1107	<code>\tabu@private@columntype</code> . . . . .	129, 136, 139
<code>\tabu@lineon</code> . . . . .	1061, 1064, 1079	<code>\tabu@privatecolumns</code> . . . . .	137, 140, 167
<code>\tabu@lines</code> . . . . .	55, 169, 248, 948	<code>\tabu@privatecolumntype</code> . . . . .	118, 141, 144, 152
<code>\tabu@lines@</code> . . . . .	949, 951, 954	<code>\tabu@pushbegins</code> . . . . .	658, 667, 668
<code>\tabu@linestyle</code> . . . . .	1128, 1129, 1138	<code>\tabu@raggedleft</code> . . . . .	312, 1625
<code>\tabu@long@arrayleft@measure</code> . . . . .	279, 628	<code>\tabu@raggedright</code> . . . . .	312, 1626
<code>\tabu@long@endTRIAL</code> . . . . .	637, 641, 655	<code>\tabu@removehfil</code> . . . . .	1492, 1498, 1504, 1521, 1528, 1535, 1567
<code>\tabu@LT@startpbox</code> . . . . .	280, 1688	<code>\tabu@restoreeverycr</code> . . . . .	910, 911
<code>\tabu@make@arstrutbox</code> . . . . .	1321, 1352, 1361	<code>\tabu@rewrite@multicolumn</code> . . . . .	246
<code>\tabu@makeatletter</code> . . . . .	1152	<code>\tabu@rewrite@X</code> . . . . .	141, 148, 449, 599, 864
<code>\tabu@maybecolor</code> . . . . .	983, 1061	<code>\tabu@rewrite@Xnested</code> . . . . .	475, 599
<code>\tabu@measure@endpbox</code> . . . . .	846	<code>\tabu@rewrite@Xrestore</code> . . . . .	148, 475
<code>\tabu@measure@startpbox</code> . . . . .	589, 685, 775, 846	<code>\tabu@rewritefirst</code> . . . . .	159, 175, 320, 368
<code>\tabu@measuringfalse</code> . . . . .	319, 649, 675	<code>\tabu@rewritefirst<sub>␣</sub>(new<sub>␣</sub>column<sub>␣</sub>type)</code> . . . . .	159
<code>\tabu@measuringtrue</code> . . . . .	456, 695, 702, 709, 768	<code>\tabu@rewritelast</code> . . . . .	193, 201, 205
<code>\tabu@message</code> 27, 34, 461, 465, 684, 692, 693, 694, 698, 738, 749, 756, 757, 760, 761, 836		<code>\tabu@rewritelast<sub>␣</sub>(new<sub>␣</sub>column<sub>␣</sub>type)</code> . . . . .	192
<code>\tabu@message@arith</code> . . . . .	684, 770	<code>\tabu@rewritemiddle</code> . . . . .	171, 174, 192, 195
<code>\tabu@message@etime</code> . . . . .	30, 792	<code>\tabu@Rlinesep</code> . . . . .	1239, 1240
<code>\tabu@message@negcoef</code> . . . . .	738, 770	<code>\tabu@row@font</code> . . . . .	1368, 1369
<code>\tabu@message@reached</code> . . . . .	692, 693, 694, 784	<code>\tabu@rowfont</code> . . . . .	54, 316, 1368
<code>\tabu@message@save</code> . . . . .	27, 35, 1338	<code>\tabu@rowfont@reset</code> . . . . .	1370, 1391
<code>\tabu@message@spread</code> . . . . .	749, 770	<code>\tabu@rowfontreset</code> . . . . .	922, 1370, 1393, 1401
<code>\tabu@minrowclearance</code> . . . . .	931, 944	<code>\tabu@Rrowsep</code> . . . . .	1227, 1228
<code>\tabu@msg@align</code> . . . . .	760, 771, 772, 773, 774, 777, 780, 781, 782, 783, 785, 786, 787, 806	<code>\tabu@sanitizetext</code> . . . . .	288, 314, 1151
<code>\tabu@multicolumn</code> . . . . .	253, 254, 258	<code>\tabu@save@decl</code> . . . . .	53, 115, 173, 250
<code>\tabu@multicolumn</code> . . . . .	56, 253, 293	<code>\tabu@savecounters</code> . . . . .	596, 609, 612
<code>\tabu@multicolumnORI</code> . . . . .	253, 257, 293	<code>\tabu@savdparam</code> . . . . .	375, 841, 1302
<code>\tabu@naturalX</code> . . . . .	87, 331, 595, 747, 752, 783, 853, 882, 883, 884	<code>\tabu@savdpream</code> . . . . .	387, 842, 1303
<code>\tabu@naturalX@max</code> 333, 586, 747, 883, 884, 906		<code>\tabu@savdpreamble</code> 178, 625, 631, 635, 1301	
<code>\tabu@naturalX@min</code> 332, 587, 748, 887, 888, 905		<code>\tabu@saveerr</code> . . . . .	145, 147, 153, 155, 1362
<code>\tabu@nbcols</code> . . . . .	81, 682, 937, 1024, 1034, 1038, 1040, 1041, 1346, 1411, 1414, 1419, 1420, 1422, 1427, 1433, 1435, 1436, 1462	<code>\tabu@savepreamble</code> . . . . .	162, 164, 176, 191
<code>\tabu@NC@list</code> . . . . .	171, 177, 193, 248, 320	<code>\tabu@savetok</code> . . . . .	1462, 1464
<code>\tabu@NC@rewrite@X</code> . . . . .	163	<code>\tabu@savetok</code> . . . . .	1425, 1427, 1430, 1434, 1436, 1461, 1463, 1476
<code>\tabu@negcoefficientfalse</code> . . . . .	319	<code>\tabu@savewarn</code> . . . . .	1298, 1299, 1362
<code>\tabu@negcoefficienttrue</code> . . . . .	213, 554	<code>\tabu@savewd</code> 843, 1306, 1307, 1345, 1349, 1351	
<code>\tabu@nested</code> . . . . .	81, 241, 289, 292, 324, 327, 335, 723, 791, 812, 900, 902, 904, 935	<code>\tabu@scandecimal</code> . . . . .	1164, 1166, 1167, 1174, 1179, 1182
<code>\tabu@new@columntype</code> . . . . .	119, 124, 127, 138	<code>\tabu@setlinesep</code> . . . . .	1233, 1235, 1238
<code>\tabu@newcolumntype</code> . . . . .	118, 159, 192, 205, 206, 246, 952	<code>\tabu@setrowsep</code> . . . . .	1221, 1223, 1226
<code>\tabu@next</code> . . . . .	208, 220, 223, 232, 235, 236, 237, 1115, 1116, 1117, 1119	<code>\tabu@settarget</code> . . . . .	274, 281, 337
<code>\tabu@noadjustspacing</code> . . . . .	867, 876, 878	<code>\tabu@setup</code> . . . . .	274, 281, 285
<code>\tabu@nocellspace</code> . . . . .	865, 874	<code>\tabu@setup@everycr</code> . . . . .	56, 215, 218, 624, 631, 635, 907
<code>\tabu@nocline</code> . . . . .	54, 61	<code>\tabu@setup@saveglobal</code> . . . . .	309, 322, 336
		<code>\tabu@setup@TRIAL</code> . . . . .	580, 594
		<code>\tabu@skipdecimal</code> . . . . .	1167, 1173, 1175, 1207
		<code>\tabu@spaces</code> . . . . .	775, 789, 792, 798, 804, 892
		<code>\tabu@spread</code> . . . . .	348, 353
		<code>\tabu@spreadarith</code> . . . . .	646, 672, 746



<code>\tabu@spreadfalse</code>	319, 768	<code>\tabu@Xfinish</code>	616, 629, 652, 678
<code>\tabu@spreadheader</code>	779, 815	<code>\tabu@xfootnote</code>	311, 402, 446
<code>\tabu@spreadtarget</code>	87, 457, 458, 470, 750, 751, 753, 754, 757, 780	<code>\tabu@xfootnotetext</code>	310, 399
<code>\tabu@spreadtrue</code>	353	<code>\tabu@xleaders</code>	1000, 1014, 1043, 1045, 1054, 1059, 1092, 1131, 1136, 1140
<code>\tabu@start</code>	81, 999, 1004, 1036, 1038, 1039	<code>\tabu@Xmath</code>	489, 528, 563, 567
<code>\tabu@start@stop</code>	1034, 1035, 1042	<code>\tabu@Xparse</code>	479
<code>\tabu@startpboxORI</code>	588, 686, 765, 847	<code>\tabu@Xparsespace</code>	494, 526
<code>\tabu@startstop</code>	998, 1034	<code>\tabu@Xtest</code>	493, 495, 532
<code>\tabu@starttime</code>	29, 794, 795, 796	<code>\tabu@Xtype</code>	515, 516, 517, 540, 544
<code>\tabu@stop</code>	81, 999, 1011, 1016, 1019, 1037, 1040, 1041	<code>\tabucline</code>	8, 54, 62, 317, 992
<code>\tabu@tabudecimal</code>	314, 1161, 1165, 1184	<code>\tabucline*</code>	9
<code>\tabu@target</code>	87, 219, 338, 354, 362, 387, 457, 459, 460, 464, 469, 581, 689, 700, 704, 722, 726, 750, 753, 754, 756, 757, 760, 761, 766, 767, 773, 777, 787, 840, 1310, 1313, 1328, 1329	<code>\tabucline@warn</code>	1056, 1060, 1075
<code>\tabu@temp</code>	197, 198, 200, 225, 227, 231, 234, 355, 357, 485, 489, 545, 549, 561, 569, 571, 734, 736, 737, 861, 866, 894, 956, 960, 966, 967, 970, 976, 986, 1108, 1110, 1111, 1112, 1113, 1115, 1116, 1166, 1169, 1170, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1203, 1204, 1213, 1214, 1215, 1623, 1624, 1625, 1626	<code>\tabucolX</code>	90, 476, 477, 484, 485, 561, 581, 696, 702, 704, 706, 708, 714, 721, 736, 748, 752, 767, 771, 782, 785, 802, 839, 1314, 1319
<code>\tabu@timerstart</code>	29, 37, 292	<code>\tabucolX@align</code>	488, 534, 536, 537, 560, 564
<code>\tabu@titles</code>	465, 812	<code>\tabucolX@do</code>	479
<code>\tabu@to</code>	347, 352	<code>\tabucolX@error</code>	90, 708
<code>\tabu@tr@cing</code>	23, 25, 39	<code>\tabucolX@rw</code>	143, 476, 477, 478, 484, 488, 541, 543, 553, 562, 566, 569, 570, 571
<code>\tabu@tracing</code>	20, 21, 22, 24	<code>\tabudecimal</code>	11, 314, 1161
<code>\tabu@tracing@save</code>	27, 819	<code>\tabudefaultrighttarget</code>	42, 46, 290, 295, 460, 462, 463, 601
<code>\tabu@TRIAL</code>	615, 633, 650, 661, 676	<code>\tabuDisableCommands</code>	613
<code>\tabu@TRIAL@hook</code>	607, 613, 614	<code>\tabular</code>	272
<code>\tabu@trivlist</code>	313, 1628	<code>\tabulinesep</code>	14, 78, 1231
<code>\tabu@usetabu</code>	1316, 1326	<code>\tabulinestyle</code>	9, 1127
<code>\tabu@vcenter</code>	106, 942	<code>\tabuphantomline</code>	7, 928
<code>\tabu@verb</code>	1152, 1158	<code>\tabusaveX</code>	838
<code>\tabu@verticalmeasure</code>	306, 605, 1254	<code>\tabustrutrule</code>	14, 93, 179, 182, 188, 1284, 1292
<code>\tabu@verticalsp@lcr</code>	1257, 1265	<code>\tabuvline@arg</code>	955, 958
<code>\tabu@verticalsp@pmb</code>	1262, 1275	<code>\tabuvline@argi</code>	970, 974, 981
<code>\tabu@verticalspacing</code>	307, 606, 1257, 1262	<code>\tabuvline@argii</code>	979, 982, 984
<code>\tabu@vline</code>	957, 962, 990	<code>\tabuvline@argiii</code>	969, 970, 985, 991
<code>\tabu@warn</code>	267, 699, 707	<code>\tabuvline@rewrite</code>	948
<code>\tabu@warn@cellspace</code>	72, 74, 80	<code>\thempfn</code>	407, 417
<code>\tabu@warn@usetabu</code>	1311, 1342	<code>\toks</code>	116, 1244, 1247, 1248, 1252, 1301, 1302, 1303, 1304, 1307, 1315, 1317, 1320, 1322, 1337, 1449, 1455, 1458, 1459, 1465, 1466
<code>\tabu@wd</code>	575, 576, 733, 736, 739, 743, 800, 858, 889, 890, 895, 896, 900, 1347, 1348	<code>\tracingtabu</code>	19
<code>\tabu@wddef</code>	834, 891, 901, 1348	<code>\tracingtabu_(debugshow_option)</code>	19
<code>\tabu@wdlet</code>	481, 556, 903	<code>\ttfamily</code>	1157
<code>\tabu@write</code>	261, 265, 269		
<code>\tabu@X@cols</code>	81, 160, 210, 330, 450, 453, 480, 481, 484, 556, 571, 583, 584, 585, 731, 732, 733, 735, 736, 738, 739, 741, 850, 851, 852, 886, 889, 890, 891, 893, 895, 896		
<code>\tabu@X@sum</code>	90, 453, 454, 467, 468, 483, 487, 549, 554, 555, 557, 561, 687, 692, 693, 696, 702, 704, 705, 710, 725, 728, 737, 774, 776, 778, 885		
<code>\tabu@Xalign</code>	518, 519, 520, 521, 522, 523, 524, 525, 533, 539		
<code>\tabu@Xarg</code>	449, 479		
<code>\tabu@Xc@ef</code>	546, 548, 551		
<code>\tabu@Xcoef</code>	527, 529, 545, 547		

## U

<code>\unhbox</code>	643
<code>\unkern</code>	1573
<code>\unless</code>	1468
<code>\unskip</code>	1156, 1578, 1678
<code>\unvbox</code>	1287
<code>\usetabu</code>	15, 144, 151, 168, 485, 1343, 1366
<code>\usetabu_(private_column_type)</code>	144

## V

<code>\value</code>	406, 415, 610
<code>\vbox</code>	934, 1260
<code>\vcenter</code>	106, 934, 942
<code>\verbatim</code>	10, 288, 314
<code>\vskip</code>	1003, 1285, 1293
<code>\vspace</code>	935
<code>\vtop</code>	898, 1260

## W

<code>\write</code>	261, 264, 268, 602
---------------------	--------------------

X	Z
<code>\X_(private_column_type)</code> . . . . .	<a href="#">141</a>
<code>\xleaders</code> . . . . .	<a href="#">1055</a> , <a href="#">1092</a>
	<code>\zap@space</code> . . . . . <a href="#">1122</a>