

# tabu

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

FC

2010/10/28 – version 1.1

### 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.

`tabu` is more flexible than `tabular`, `tabular*`, `tabularx` and `array` and extend the possibilities. All tabulars in this document were made with the `tabu` environment. Last (but not least) `tabu` is more than compatible with any package that provide commands to format tabulars. Indeed, **tabu does not modify any of the macros of `array.sty`**. All functionalities are obtained with only two modifications<sup>1</sup> which are loaded only inside a `tabu` environment.

`tabu` requires  $\epsilon$ -T<sub>E</sub>X, the standard package `array.sty` and no other package.

### Contents

<b>1</b>	<b>The <code>tabu</code> environment</b>	<b>3</b>
1.1	<code>tabu</code> , <code>tabu to</code> and <code>tabu spread</code>	3
1.2	<code>tabu X</code> columns	3
1.2.1	<i>X columns with “<code>tabu spread</code>”</i>	
1.2.2	<i>Nesting <code>tabus</code> with X columns</i>	
1.3	Lines inside <code>tabu</code>	5
1.3.1	<i>First important remarks</i>	
1.3.2	<i>Vertical lines: <code> </code> has an optional parameter</i>	
1.3.3	<i>More style for horizontal lines: <code>\tabucline</code></i>	
1.4	<code>tabu</code> in math mode	7
1.5	Verbatim inside <code>tabu</code> with X columns	8
1.6	Printing numbers inside <code>tabu</code> with <code>numprint</code> and <code>siunitx</code>	8
1.6.1	<i>Just make it easy !</i>	
1.6.2	<i>You should know how it works...</i>	
1.7	<code>delarray</code> shortcuts	9
<b>2</b>	<b>Save the <code>tabu</code> preamble and X columns widths: <code>\savetabu</code>, <code>\usetabu</code></b>	<b>9</b>
<b>3</b>	<b>Change the font and the alignment in one row: <code>\rowfont</code></b>	<b>10</b>
<b>4</b>	<b>Differences between <code>tabu</code>, <code>tabular</code> and <code>tabularx</code></b>	<b>11</b>
4.1	Paragraph indentation	11
4.2	Custom environments	11
4.3	Inversion of tokens	11
4.4	Improved process for rewriting columns	12
<b>5</b>	<b>The package options</b>	<b>12</b>
5.1	The <code>debugshow</code> package option	12
5.2	The <code>linegoal</code> package option	13
5.3	The <code>delarray</code> package option	13
5.4	The <code>light</code> package option	13
<b>6</b>	<b>Corrections of some bugs (available only inside <code>tabu</code>)</b>	<b>13</b>
6.1	Correction for <code>colortbl</code> and <code>arydshln</code> : compatibility with <code>delarray</code>	13
6.2	Correction for <code>arydshln</code> : <code>@</code> columns	14

This documentation is produced with the +DocStrip+ utility. Package `tabu` is used with the `linegoal` option.

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

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

1. `\prepnext@tok` and `\save@decl`. See “[The light package option](#)”.

<b>7</b>	<b>IMPLEMENTATION</b>	<b>15</b>
7.1	Identification, requirements and options	15
7.2	Some constants	16
7.3	<code>\tabu</code> and <code>\endtabu</code>	19
7.4	Flow chart of expansion	21
7.4.1	General case: <code>tabu</code> , <code>tabu to</code> and <code>tabu spread</code>	
7.4.2	<code>tabu to</code> with <i>X</i> column	
7.4.3	<code>tabu spread</code> with <i>X</i> column	
7.5	<code>tabu X</code> column definition	23
7.6	Before trials to reach the target	25
7.6.1	Trial setup after <code>\@mkpream</code>	
7.6.2	Collecting the <code>tabu</code> body: required for <i>X</i> columns	
7.7	Measuring the <code>tabu</code>	27
7.7.1	One trial after another	
7.7.2	The arithmetic of <i>X</i> columns	
7.7.3	Reporting in the <code>.log</code> file ( <code>debugshow</code> option)	
7.7.4	Measuring the natural width for <code>tabu spread</code>	
7.8	Numbers in <code>tabu</code>	29
7.8.1	<code>\tabudecimal</code>	
7.9	Lines inside <code>tabu</code>	31
7.9.1	Vertical lines	
7.9.2	Horizontal lines: <code>\tabucline</code>	
7.10	Verbatim inside <code>tabu</code> with <i>X</i> columns	35
7.11	<code>\savetabu</code>	35
7.12	<code>\rowfont</code>	36
7.12.1	Setting font and alignment specification	
7.12.2	Preparing stuff to be able to use <code>\rowfont</code>	
7.12.3	Neutralisation of glues and alignment modification	
7.13	Utilities	41
7.13.1	<code>tabu \fbox</code>	
7.13.2	<code>\centering</code> , <code>\raggedright</code> , <code>\raggedleft</code>	
7.14	Corrections	41
7.14.1	<code>delarray</code> compatibility fix for <code>colortbl</code> and <code>arydshln</code>	
7.14.2	<code>arydshln @ columns</code>	
7.14.3	<code>arydshln</code> and empty <i>p</i> columns	
<b>8</b>	<b>History</b>	<b>43</b>
	[2010/10/28 v1.0]	43
<b>9</b>	<b>Index</b>	<b>43</b>

---

# 1 The tabu environment

## 1.1 tabu, tabu to and tabu spread

```
\begin{tabu}[\langle pos \rangle]{\langle tabular preamble \rangle}
\begin{tabu} to \langle dimen \rangle[\langle pos \rangle]{\langle tabular preamble \rangle}
\begin{tabu} spread \langle dimen \rangle[\langle pos \rangle]{\langle tabular preamble \rangle}
```

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

- footnotes and index words are allowed inside `tabu`
- X columns are also allowed, with an *optional* parameter for the width-coefficient, the alignment (l,r,c or L,R,C for `ragged2e` settings) and the column type (p,m,b). `tabu` has also a *default target width* when used with X columns.
- 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` introduces optional parameters to columns specifications, providing facilities for modifying columns of line widths, inserting verbatim text...
- `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 \langle dimen \rangle` 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 you specify another value for `\extracolsep`. Therefore “`tabu to`” fills in width the specified `\langle dimen \rangle`.

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

## 1.2 tabu X columns

`tabu` X columns can be viewed as an enhancement of `tabularx` X columns:

- 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
--------	---

- 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	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` without any restriction.
- `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 optimized: 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 pdfT<sub>E</sub>X either in `pdf` or `dvi` mode, and package `linegoal` is loaded. Remember that `\linegoal` uses `zref-savepos` and `\pdfsavepos`: 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 is 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|}
```

<p>"Like the air we breathe, Sherlock Holmes is everywhere. His pipe-smoking, deer stalkered 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 to become an international best-seller and has been accepted as part of British folklore. Holmes is alive to millions."</p>	<p>There the text was too long, and <code>tabu spread</code> behaves as if you didn't give it a target.</p> <p>The result of this example is the same as if one had written <code>\begin{tabu}to\linewidth.</code></p>	Sherlock Holmes
<p>And a <code>\multicolumn</code> cell, just to see</p>		

In the preamble, `@{}` means that you remove the margin.

Nesting `tabus` with `X` columns

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

1.3 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 (sorry for that, but I was tired to parse parameters... as for `X` columns, commas are optional). Example:

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

The `\verbatim` command available inside `tabu` with `X` columns was used: guess in which cells...?

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.

<div>[1pt on 1.5pt off 2pt]</div> <div>[1.5pt]</div> <div>default</div> <div>[on 2pt red]</div>	<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 gaps width is 4pt (the default).

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

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

Hello	world
Hello	world

`\tabucline`

Hello	world
Hello	world

`\tabucline*`

Hello	world
Hello	world

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

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


`\tabuclines` tries to put the line in the middle, so that the text is centered in its cell. `\tabuclines*` tries to take care of vertical lines. Well this is not perfect: it works pretty well with simple vertical lines, but not really with double lines, triple lines with colors 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}</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
And below is the default

Dash	Gap
	
This one was thick	

Dash	Gap...	...or leader
		
Interesting ?		

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

1.4 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[r$]}`.

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

```
\begin{tabu}(>\Gape[8pt]\displaystyle)r>\displaystyle{l}.
```

You immediately see the interest for `tabu` to have a default target.

`\Gape` is a `makecell` macro to adjust the height and depth of the rows.

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

array		tabu		tabu spread 1em	
$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$
$\gamma$	$\delta$	$\gamma$	$\delta$	$\gamma$	$\delta$

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

Vertical lines inside the tabular preamble gives:

$\alpha$	$\beta$
$\gamma$	$\delta$

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



## 1.5 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{\textit{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`.

## 1.6 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{\textit{user-macro}}`

`\tabudecimal` can be used in the preamble of a `tabu` before a column specification. The `\textit{user-macro}` is a macro with one parameter that have to be defined before.

Example with `\numprint`:

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

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

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

Example with `\SI`:

```
\def\usermacro#1{\SI[group-four-digits=true, % thousand separator
```



```

round-mode=places,           % round numbers
round-precision=2,           % with 2 decimal digits
round-integer-to-decimal=true, % add trailing 0 if necessary
per-mode=symbol]{#1}{\officialeuro\per\kilo\gram}}
\begin{tabu}spread Opt{ |[GreenYellow]*2{>{\tabudecimal \usermacro}X[r]}|[GreenYellow]}} ...

```

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

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

### 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...

## 1.7 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:

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

With overfull hboxes (15.8pt and 16.1pt too wide) for **tabularx**.

## 2 Save the **tabu** preamble and **X** columns widths: `\savetabu`, `\usetabu`

```
\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.

`\usetabu{⟨user-name⟩}`

`\usetabu` is the complement of `\savetabu`: it can be used 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` locally restores:

- the preamble,
- 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, and `X` columns are directly transformed into `p`, `m` or `b` columns.
- `\tabcolsep` (or `\arraycolsep` in math mode)
- `\arrayrulewidth`, `\doublerulesep`, `\extrarowheight` and `\extratabsurround`

Example:

```
\extrarowheight=5pt\tabcolsep=12pt
\begin{tabu}to .7\linewidth{|XXX|X[c]|}\savetabu{mytabu} \tabucline*[on1pt]-
  This & is & tabu & package \\ \tabucline*[on1pt]-
\end{tabu}
\begin{tabu}{\usetabu{mytabu}} \tabucline*[on1pt]-
\multicolumn{3}{|c|}{This is tabu} & package \\ \tabucline*[on1pt]-
\end{tabu}
```

This	is	tabu	package
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. `\usetabu` is a help to get several tabulars of exactly the same shape, same target, same preamble.

### 3 Change 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:

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

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.

## 4 Differences between tabu, tabular and tabularx

### 4.1 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`

### 4.2 Custom environments

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

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

`tabu` environment, even when `X` columns are used, may appear in the definition of your custom `tabular` environment.

You can also use the commands `\savetabu` and `\usetabu` for this purpose.

### 4.3 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.

## 4.4 Improved process for rewriting columns

In the following example, you get an error with `tabular` and no error with `tabu`. With `tabular`, and `siunitx` `S` column, the rewriting 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> 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, not rebuilt before each trial as `tabularx` does...

The process of rewriting is very sensitiv to the order in which columns are actually rewritten. Therefore, if it possible to define a new column type 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 type is defined before the `C` column type...

Well, just avoid to nest new column type definitions !

## 5 The package options

### 5.1 The `debugshow` package option

`\tracingtabu`

With the package option `debugshow`, `tabu` will report the widths it computes at each attempt to read the target, when `X` columns are used. The control sequence `\tracingtabu` has the same effect as the `debugshow` option.

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  $386.67294\text{pt}$ : the target is reached.  
The final width of each  $X$  column is the product of `tabu`  $X$  by its width coefficient.

## 5.2 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\tabudefaultright=\linewidth`

## 5.3 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.

## 5.4 The `light` package option

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

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

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

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

If for some reason you prefer the original macros of `array.sty`, then you can load `tabu` with the “light” option. As a consequence: `\rowfont` and `\tabucline` will not be available, and vertical lines `|` will not have an optional argument for their widths and color. However, `tabu`  $X$  columns will work normally.

# 6 Corrections of some bugs (available only inside `tabu`)

## 6.1 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`.

Those control sequences are used by the `delarray` package to put parenthesis or bracket around the array.

## 6.2 Correction for arydshln: @ columns

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

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

## 7 Implementation

### 7.1 Identification, requirements and options

The package namespace is `tabu@`.

```

1 <*package>
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{tabu}[2010/10/28 v1.1 - flexible LaTeX tabulars (FC)]
4 \RequirePackage{array}[2008/09/09]
```

Then minimal catcode ascertaining for loading `tabu` in good conditions:

```

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

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

```

18 \let\tabu@message\@gobble
19 \def\tracingtabu{\let\tabu@message\message}
20 \let\tabudefaulttarget\linewidth
21 \DeclareOption{debugshow}{\tracingtabu}
```

`linegoal` (package option)

```

22 \DeclareOption{linegoal}{%
23   \AtEndOfPackage{\RequirePackage{linegoal}[2010/10/31]}%
24   \def\tabudefaulttarget{\linegoal}% \linegoal is \linewidth if not pdfTeX
25 }% linegoal option
```

`delarray` (package option)

```

26 \DeclareOption{delarray}{%
27   \AtEndOfPackage{\RequirePackage{delarray}}%
28 }% delarray option
```

`\light` (package option)

```

29 \DeclareOption{light}{%
30   \AtEndOfPackage{%
31     \let\tabu@prepnext@tok \prepnext@tok
32     \let\tabu@save@decl \save@decl
33     \let\tabu@rowfont \tabu@norowfont
34     \let\tabucline \tabu@nocline
35     \let\tabu@firstcline \relax
36     \let\tabu@lines \relax
37   }
38 }% light option
39 \def\tabu@norowfont{\PackageError{tabu}
40   {\string\rowfont\space is not available with option 'light'}\@ehd}
41 \def\tabu@nocline{\PackageError{tabu}
42   {\string\tabucline\space is not available with option 'light'}\@ehd}
```



```
43 \ProcessOptions
```

At Begin Document, we check if a **X** column has already been defined (`tabularx`) and if not, we define a new column type **X**.

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

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

```
44 \AtBeginDocument{%
45   \@ifundefined{NC@rewrite@X}{\newcolumnntype{X}{}}{}% new column X if not exists
46   \expandafter\in@\expandafter\@arrayright\expandafter{\endarray}%
47   \ifin@ \let\tabu@endarray\endarray
48   \else \tabu@fix@arrayright \fi % <fix for colortbl & arydshln (delarray)>
49   \@ifpackageloaded{colortbl} \tabu@colortbltrue \tabu@colortblfalse
50 }
```

## 7.2 Some constants

<code>\tabu@cnt</code>	Used in 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> ”.
	<pre>51 \newcount\tabu@cnt 52 \newcount\tabu@nbcols 53 \newcount\tabu@X@cols</pre>
<code>\tabu@target</code>	Stores the <code>tabu</code> target (either “ <code>to</code> ” or “ <code>spread</code> ”).
<code>\tabu@spreadtarget</code>	Used only when <code>tabu X</code> columns are used with “ <code>tabu spread</code> ”.
<code>\tabu@naturalX</code>	<pre>54 \newdimen\tabu@target 55 \newdimen\tabu@spreadtarget 56 \newdimen\tabu@naturalX</pre>
<code>\tabucolX</code>	The <code>dimen</code> corresponding to the preamble token <code>X[1]</code> : the standard width of <b>X</b> columns.
<code>\tabu@X@sum</code>	Stores the sum of all width coefficients for <b>X</b> columns.
	<pre>57 \newdimen\tabucolX 58 \newdimen\tabu@X@sum</pre>
<code>\iftabu@measuring</code>	This switch is set to <code>true</code> by <code>\tabu@arith</code> if the trial did not reached the target. It is also temporarily set to <code>true</code> when the first <b>X</b> column is encountered in the <code>tabu</code> preamble, at the time <code>\@mkpream</code> scans it to built the <code>\halign</code> preamble. The first <b>X</b> column found actually triggers some special setup to be expanded before <code>\halign</code> (see the flow chart...)
<code>\iftabu@spread</code>	A switch whether “ <code>tabu spread</code> ” is used or not.
<code>\iftabu@nested</code>	A switch set at the entry in <code>tabu</code> environment: <code>true</code> if <code>tabu</code> is nested inside another <code>tabu</code> .
<code>\iftabu@firstcline</code>	A switch to adapt <code>\tabucline</code> automatically if it comes first in the <code>tabu</code> (similar to <code>\firsthline</code> ).
	<pre>59 \newif\iftabu@measuring 60 \newif\iftabu@spread 61 \newif\iftabu@nested 62 \newif\iftabu@firstcline</pre>
<code>\tabu@box</code>	Stores the whole <code>tabu</code> when an attempt to adjust <b>X</b> columns is performed. It is also used by <code>\tabucline</code> to save the <code>\@arstrutbox</code> when inserting a horizontal line.
	<pre>63 \newsavebox\tabu@box</pre>

`\tabu@gobblespace` A macro which is needed when scanning tokens with `\futurelet`.

```
64 \def\tabu@gobblespace#1 {#1}
```

`\tabu@NC@rewrite@X` 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}`

```
65 \newcommand\tabu@NC@rewrite@X[1] [] {\tabu@rewrite@X{#1}%
66 \expandafter \NC@find \tabucolX@spec}
```

The next part of the definition (`\tabu@rewrite@X`) can be found page 23.

`\usetabu (new column type)` `\usetabu` is defined as a new column type. `\NC@rewrite@usetabu` is expanded where `\@mkpream` does its job.

```
67 \expandafter\def\csname NC@rewrite@\string\usetabu\endcsname#1{%
68 \ifx\#1\\\tabu@saveerr{}\else
69 \ifundefined\tabu@saved@\string#1}
70 {\tabu@saveerr{#1}}
71 {\let\tabu@rewrite@X \tabu@rewrite@Xrestore
72 \def\tabu@temp{\xdef\tabu@usetabu{%
73 \col@sep \the\col@sep\relax
74 \arrayrulewidth \the\arrayrulewidth\relax
75 \@tempdima \ht\@arstrutbox \advance\@tempdima -\extrarowheight
76 \extrarowheight \the\extrarowheight\relax
77 \advance\@tempdima \extrarowheight
78 \ht\@arstrutbox \@tempdima
79 \extratabsurround \the\extratabsurround\relax
80 \doublerulesep \the\doublerulesep\relax}}%
81 \aftergroup \tabu@usetabu
82 \csname tabu@saved@\string#1\expandafter\endcsname}%
83 \fi
84 }% \NC@rewrite@usetabu
85 \expandafter\def\csname NC@find@\string\usetabu\endcsname#1\usetabu{\NC@{#1}}
```

`\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).

```
86 \expandafter\def\csname NC@rewrite@\string\tabu@rewritefirst\endcsname{%
87 \ifx\tabu@lines\relax
88 \NC@list\expandafter{\expandafter\NC@do \expandafter\usetabu
89 \NC@do X\NC@do\tabu@rewritelast}%
90 \else
91 \NC@list\expandafter{\expandafter\NC@do \expandafter\usetabu
92 \expandafter\NC@do \expandafter|\tabu@NC@list
93 \NC@do X\NC@do\tabu@rewritelast}%
94 \tabu@lines % defines NC@rewrite@| for tabu only (inside @mkpream group)
95 \fi
96 \let\save@decl \tabu@save@decl % inversion of tokens in text mode
97 \let\NC@rewrite@X \tabu@NC@rewrite@X
98 \aftergroup \tabu@global@temp
99 \aftergroup \tabu@firstcline
100 \NC@find \tabu@rewritelast
101 }% \NC@rewrite@\tabu@rewritefirst
102 \expandafter\def\csname NC@find@\string\tabu@rewritefirst\endcsname
```

```

103                                     #1\tabu@rewritefirst{\NC@{#1}}
104 \def\tabu@rewritefirst{%
105   \edef\tabu@NC@list{\the\NC@list}%
106   \NC@list{\NC@do \tabu@rewritefirst \NC@do *}%
107 }% \tabu@rewritefirst

```

`\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.

```

108 \expandafter\def\csname NC@rewrite@\string\tabu@rewritelast\endcsname{%
109   \global\NC@list\expandafter{\tabu@NC@list}%
110   \futurelet \tabu@temp \tabu@rewritelast
111 }% \NC@rewrite@\tabu@rewritelast
112 \expandafter\def\csname NC@find@\string\tabu@rewritelast\endcsname
113                                     #1\tabu@rewritelast{\NC@{#1}}
114 \def\tabu@rewritelast{%
115   \ifx \@halignto\relax \let\tabu@temp \@empty
116   \else
117     \let\@halignto \tabu@halignto
118     \ifdim \tabu@target=z@ \let\tabu@temp \@empty
119     \else
120       \if |\noexpand\tabu@temp \def\tabu@temp{@{\extracolsep\@flushglue}}\else
121       \if !\noexpand\tabu@temp \def\tabu@temp{@{\extracolsep\@flushglue}}\else
122                                     \def\tabu@temp{!\extracolsep\@flushglue}\fi\fi
123     \fi
124   \fi
125   \expandafter\NC@find \tabu@temp
126 }% \tabu@rewritelast

```

`\tabu@everycr@tok` This token is used to store and restore the content of `\everycr` when `\rowfont` is used.

`\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.

```

127 \newtoks\tabu@everycr@tok
128 \newif\iftabu@colortbl

```

`\tabu@nowrite` A trick to fobidd `\write` when a trial is done on the `\halign`. (Copied from D. Carlisle’s `tabularx` code.)

```

129 \def\tabu@nowrite{% cancels a \write command (tabularx method)
130   \begingroup
131     \def\let{\afterassignment\endgroup\toks@}%
132     \afterassignment\let\count@
133 }% \tabu@nowrite
134 \let\tabu@write\write

```

`\tabu@phantomline` 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...`

```

135 \def\tabu@phantomline{\noalign{%
136   \global\everycr{}%
137   \global\setbox\tabu@box\box\@arstrutbox
138   \toks@{}\count@\@ne
139   \@whilenum\count@<\tabu@nbcols\do{\advance\count@\@ne

```

```

140             \toks@\expandafter{\the\toks@&}}%
141   \toks@\expandafter{\the\toks@
142     \cr\noalign{\global\setbox\@arstrutbox\box\tabu@box}}%
143   \expandafter\the\toks@
144 }% \tabu@phantomline

```

### 7.3 \tabu and \endtabu

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

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

```

145 \def\tabu{%
146   \ifmmode \def\endtabu{\endarray}%
147   \else    \def\endtabu{\endtabular}\fi
148   \tabu@setup \tabu@settarget
149 }% \tabu

```

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

Incidentally, `\tabu@X` (number of `tabu` X columns) and `\tabu@X@sum` (sum of the width-coefs) are reset to 0.

The current value of `\hfuzz` is stored in `\tabu@hfuzz`, with a minimum of 0.1pt.

```

150 \def\tabu@setup{%
151   \let\adl@xarraydashrule \tabu@adl@xarraydashrule % <fix> arydshln
152   \let\adl@act@endpbox \tabu@adl@act@endpbox % <fix> arydshln
153   \let\adl@act@@endpbox \tabu@adl@act@endpbox % <fix> arydshln
154   \let\endarray \tabu@endarray % <fix> colortbl & arydshln (delarray)
155   \let\tabu@global@temp \@empty \let\tabu@global@X \@empty
156   \ifx\verbatim \tabu@sanitizetext
157     \tabu@nestedtrue
158     \def\tabu@aligndefault{t}\def\tabu@defaulttarget{\linewidth}%
159   \else \tabu@nestedfalse
160     \def\tabu@aligndefault{c}%
161     \ifdim\parindent>\z@ \ifx\linewidth\tabu@defaulttarget
162       \everypar\expandafter{% % correction for indentation
163         \the\everypar\everypar\expandafter{\the\everypar}%
164         \setbox\z@=\lastbox
165         \ifdim\wd\z@>\z@ \advance\linewidth -\wd\z@\fi
166         \box\z@
167       }%
168     \fi\fi
169   \fi
170   \let\@footnotetext \tabu@footnotetext
171   \let\@xfootnotenext\tabu@xfootnotetext
172   \iftabu@nested\else
173     \global\tabu@footnotes{}%
174     \aftergroup\the\aftergroup\tabu@footnotes
175   \fi
176   \let\centering \tabu@centering
177   \let\raggedright \tabu@raggedright
178   \let\raggedleft \tabu@raggedleft
179   \let\@trivlist \tabu@trivlist %<restore \\=\@centercr inside lists>
180   \let\tabu@decimal \tabu@tabudecimal
181   \let\verbatim \tabu@sanitizetext
182   \let\fbbox \tabu@fbbox
183   \let\prepnext@tok \tabu@prepnext@tok % <for rowfont and tabucline>
184   \let\rowfont \tabu@rowfont
185   \tabu@spreadfalse \tabu@measuringfalse

```

```

186 \edef\tabu@hfuzz{\ifdim\hfuzz<.1\p@ .1\p@\else\the\hfuzz\fi}%
187 \tabu@rewritefirst
188 }% \tabu@setup
189 \def\tabu@save@decl{% no inversion on tokens when not in math mode
190 \ifcat$\d@llarend
191 \toks\count@\expandafter\expandafter\expandafter{%
192 \expandafter\@nextchar \the\toks\count@}%
193 \else
194 \toks\count@\expandafter\expandafter\expandafter{%
195 \expandafter\the\expandafter\toks\expandafter\count@\@nextchar}%
196 \fi
197 }% \tabu@save@decl

```

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

```

\tabu@begin
198 \def\tabu@settarget{\futurelet\@let@token \tabu@settargetget}
199 \def\tabu@settargetget{\tabu@target\z@
200 \ifcase \ifx \bgroup\@let@token 0\else
201 \ifx [\@let@token 0\else
202 \ifx \@sptoken\@let@token 1\else
203 \if t\@let@token 2\else
204 \if s\@let@token 3\else
205 \m@ne\fi\fi\fi\fi\fi\relax
206 \expandafter\tabu@begin
207 \or \expandafter\tabu@gobblespace\expandafter\tabu@settarget
208 \or \expandafter\tabu@to
209 \or \expandafter\tabu@spread
210 \else\expandafter\tabu@begin
211 \fi
212 }% \tabu@settargetget
213 \def\tabu@to to{\def\@halignto{to}\tabu@gettarget}
214 \def\tabu@spread spread{\tabu@spreadtrue\def\@halignto{spread}\tabu@gettarget}
215 \def\tabu@gettarget{\afterassignment\tabu@begin\tabu@target}
216 \def\tabu@begin#1#{%
217 \edef\@halignto{\ifdim\tabu@target>\z@ \@halignto\the\tabu@target\fi}%
218 \let\tabu@halignto \@halignto
219 \expandafter\@testopt\expandafter\tabu@@begin \tabu@aligndefault #1\@nil
220 }% \tabu@begin
221 \def\tabu@@begin[#1]#2\@nil#3{%
222 \edef\tabu@align{#1}%
223 \edef\tabu@saved{%
224 \ifmmode \col@sep \the\arraycolsep
225 \else \col@sep \the\tabcolsep \fi\relax
226 \arrayrulewidth \the\arrayrulewidth\relax
227 \extrarowheight \the\extrarowheight\relax
228 \extratabsurround \the\extratabsurround\relax
229 \doublerulesep \the\doublerulesep\relax}%
230 \expandafter\def\expandafter\tabu@saved\expandafter{\tabu@saved
231 \tabu@temp \edef\tabu@halignto{to\the\tabu@target}\NC@find #3}%
232 \ifmmode \expandafter\array
233 \else \expandafter\tabular
234 \fi [{#1}]#2{\tabu@rewritefirst #3}%
235 }% \tabu@@begin

```

`\tabu@footnotes`

```

236 \newtoks\tabu@footnotes
237 \long\def\tabu@footnotetext#1{%
238 \edef\@tempa{\the\tabu@footnotes
239 \noexpand\footnotetext [\the\csname c@\mpfn\endcsname]}%

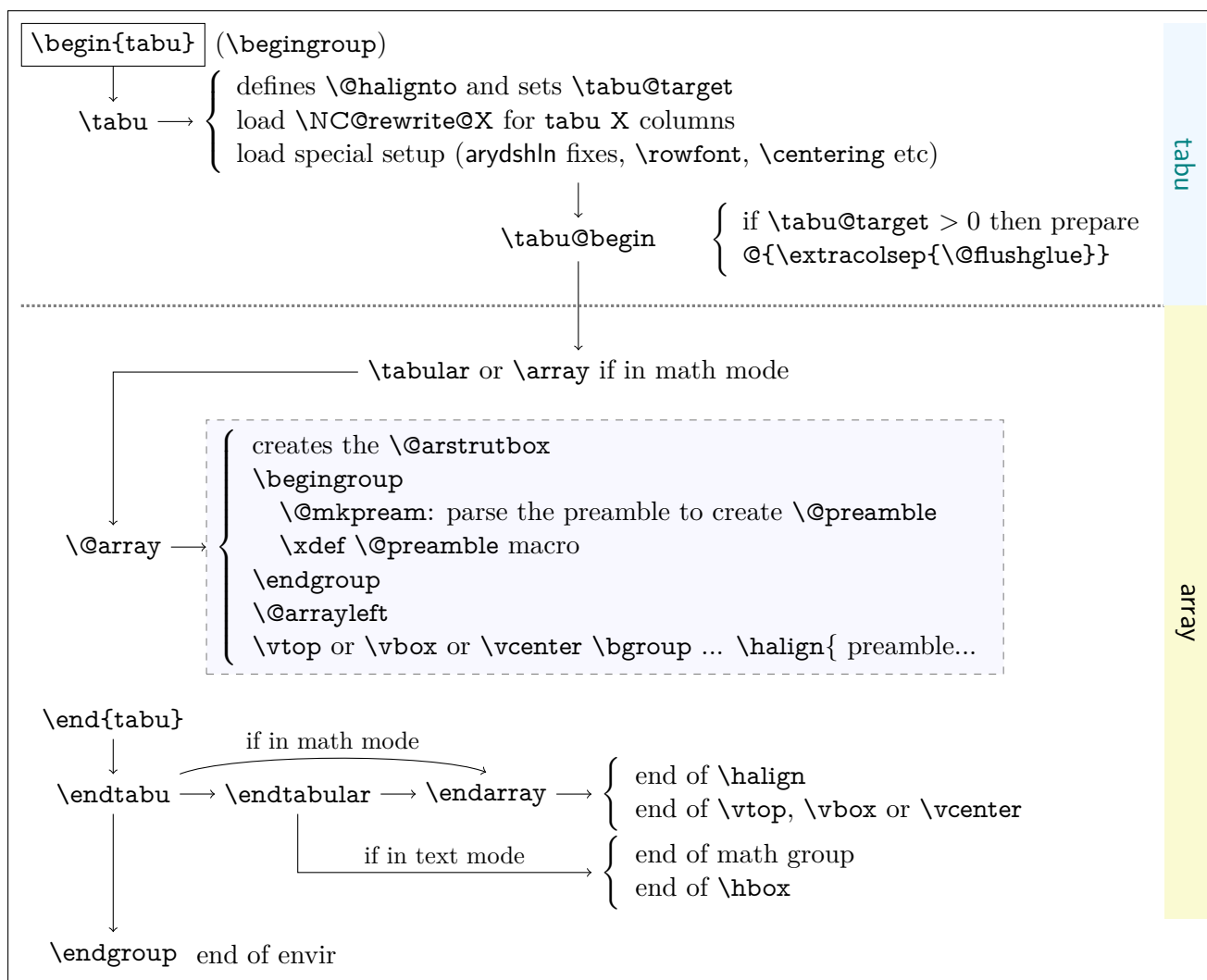
```

```

240 \global\tabu@footnotes\expandafter{\@tempa{#1}}}%
241 \long\def\tabu@xfootnotetext[#1]#2{%
242 \global\tabu@footnotes\expandafter{\the\tabu@footnotes
243 \footnotetext[{#1}]{#2}}}
```

## 7.4 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: `\@mkpream` 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`.

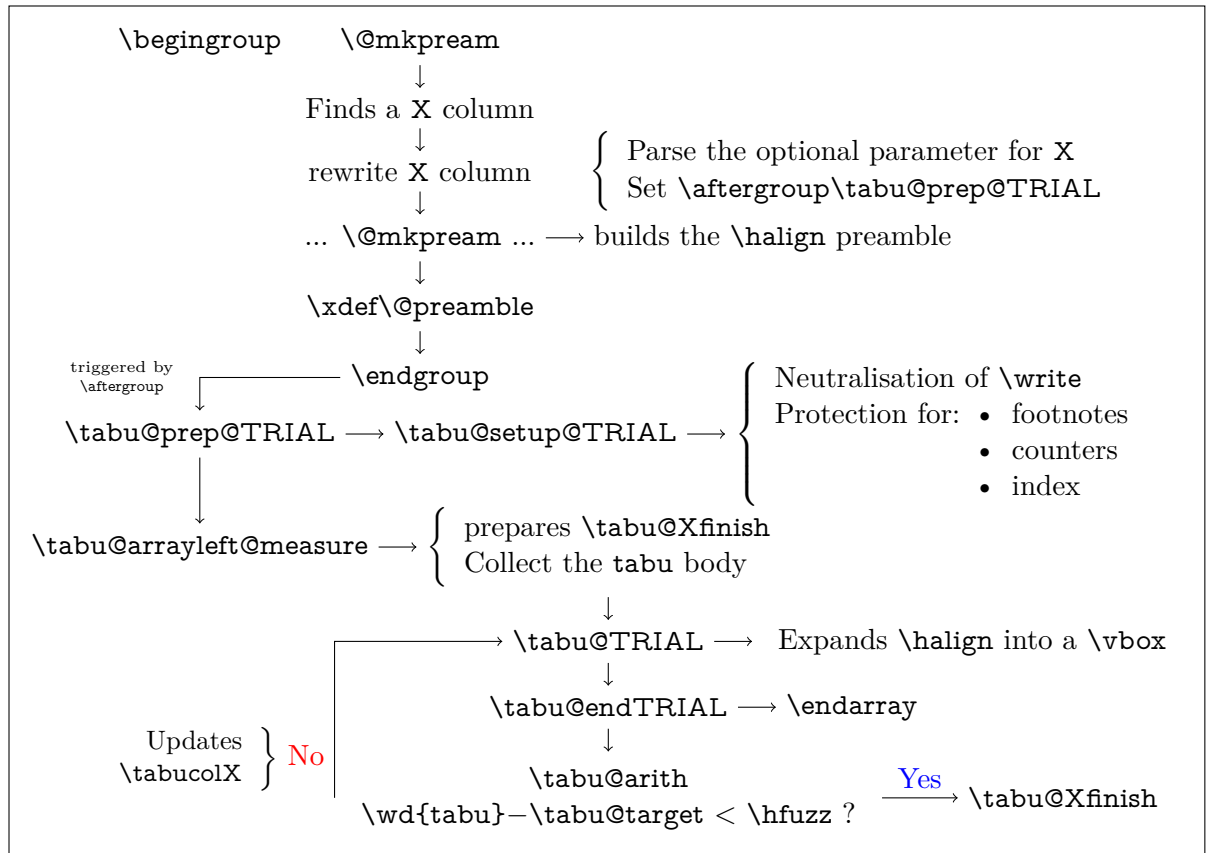
When a `tabu X` column is found in the preamble by `\@mkpream`, `tabu` changes his strategy: the macro `\tabu@prep@TRIAL` is set to be expanded `\aftergroup` that is, just after the preamble (`\@preamble`) has been built. This macro does some setup for `tabu` trials to reach the target with variable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

## 7.5 tabu X column definition

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

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

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

```
244 \def\tabu@rewrite@X#1{\tabu@Xarg{#1}%
245   \iftabu@spread \tabu@rewrite@Xspread
246   \else          \tabu@rewrite@Xto
247   \fi
248 }% \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`.

```
249 \def\tabu@rewrite@Xnested#1{\def\tabucolX@spec{p{\tabucolX}}
250 \def\tabu@rewrite@Xrestore#1{\tabu@Xarg{#1}\let\tabucolX@spec\tabu@temp
251   \iftabu@measuring \else \tabu@measuringtrue \aftergroup\tabu@usetabuX \fi}%
252 \def\tabu@usetabuX{%
253   \tabu@everycr\expandafter{\the\tabu@everycr\tabu@phantomline}%
254 }% \tabu@usetabuX
```

`\tabu@rewrite@Xto` The setup for trial is not the same in case of “tabu to” and “tabu spread”.

The important thing is: `\aftergroup\tabu@prep@TRIAL`.

```
255 \def\tabu@rewrite@Xto{%
256   \iftabu@measuring % not the first X column found in preamble
257     \xdef\tabu@global@X {\tabu@global@X
258       \advance\tabu@X@sum \the\tabu@X@sum\relax}%
259   \else % first X column found in preamble
260     \tabu@measuringtrue
261     \ifdim\tabu@target=\z@
262       \setlength\tabu@target\tabudefaulttarget
263     \fi
264     \xdef\tabu@global@X{%
265       \tabu@X@sum \the\tabu@X@sum\relax
266       \tabu@target \the\tabu@target\relax}%
267     \let \@halign \relax
268     \aftergroup \tabu@prep@TRIAL
269   \fi
270 }% \tabu@rewrite@Xto
```

`\tabu@rewrite@Xspread` This macro is used instead of `\tabu@rewrite@Xto` for X columns with “tabu spread”.

```
271 \def\tabu@rewrite@Xspread{% tabu spread with X columns: we need to store each coef
272   \iftabu@measuring % not the first X column found in preamble
273     \advance\tabu@X@cols \@ne
274     \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \relax
275     \xdef\tabu@global@X {\tabu@global@X
276       \advance\tabu@X@cols \@ne
277       \advance\tabu@X@sum \the\tabu@X@sum\relax
```

```

278         \def\csname tabu@X\the\tabu@X@cols\endcsname{\strip@pt\tabu@X@sum}}%
279     \else % first X column found in preamble
280         \tabu@measuringtrue
281         \tabu@X@cols \@ne \tabu@spreadtarget=\tabu@target
282         \setlength \tabu@target \tabu@defaulttarget
283         \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \relax
284         \xdef\tabu@global@X {%
285             \tabu@X@cols \@ne
286             \tabu@X@sum \the\tabu@X@sum\relax
287             \tabu@target \the\tabu@target\relax
288             \tabu@spreadtarget \the\tabu@spreadtarget\relax
289             \def\csname tabu@X\the\tabu@X@cols\endcsname{\strip@pt\tabu@X@sum}}%
290         \let\@halignto \relax
291         \aftergroup \tabu@prep@TRIAL
292     \fi
293 }% \tabu@rewrite@Xspread

```

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

`>\{alignment\} p or m or b {\dimexpr coef \tabucolX\relax}`

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

```

294 \def\tabu@Xarg#1{%
295     \ifx\#1\% <shortcut when no option>
296         \tabu@X@sum \p@
297         \def\tabucolX@spec{p{\tabucolX}}%
298         \edef\tabu@temp{p{\the\tabucolX}}% <required for \usetabu>
299     \else
300         \tabu@X@sum \z@
301         \let\tabucolX@align \@empty
302         \let\tabucolX@spec \@empty
303         \let\tabu@Xmath \relax
304         \tabu@Xparse {}#1\relax\@nnil
305     \fi
306 }% \tabu@Xarg
307 \def\tabu@Xparse#1{\futurelet\@let@token\tabu@Xtest}
308 \expandafter\def\expandafter\tabu@Xparsespace\space{\tabu@Xparse{}}
309 \def\tabu@Xtest{%
310     \ifcase \ifx \@nnil\@let@token \z@ \else
311         \ifx \relax\@let@token \m@ne\else
312         \if ,\@let@token \m@ne\else
313         \if p\@let@token 1\else
314         \if m\@let@token 2\else
315         \if b\@let@token 3\else
316         \if l\@let@token 4\else
317         \if c\@let@token 5\else
318         \if r\@let@token 6\else
319         \if .\@let@token 7\else
320         \ifx \@sptoken\@let@token 8\else
321         \if L\@let@token 9\else
322         \if C\@let@token 10\else
323         \if R\@let@token 11\else
324         \ifcat $\@let@token 12\else
325             13\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\relax
326         \tabucolX@do \expandafter\remove@to@nnil
327     \or \tabu@Xcolspec{p}%
328     \or \tabu@Xcolspec{m}%
329     \or \tabu@Xcolspec{b}%

```

```

330 \or \tabu@Xalign{>{\raggedright}}%
331 \or \tabu@Xalign{>{\centering}}%
332 \or \tabu@Xalign{>{\raggedleft}}%
333 \or \expandafter\tabu@Xcoef
334 \or \expandafter\tabu@Xparsespace
335 \or \tabu@Xalign{>{\RaggedRight}}%
336 \or \tabu@Xalign{>{\Centering}}%
337 \or \tabu@Xalign{>{\RaggedLeft}}%
338 \or \let\tabu@Xmath=$\expandafter\tabu@Xparse
339 \or \tabu@Xcoef{}%
340 \else \expandafter\tabu@Xparse
341 \fi
342 }% \tabu@Xtest
343 \def\tabu@Xalign#1{%
344 \ifx \tabucolX@align\@empty \else \PackageWarning{tabu}
345 {Duplicate horizontal alignment specification}\fi
346 \def\tabucolX@align{#1}\expandafter\tabu@Xparse
347 }% \tabu@Xalign
348 \def\tabu@Xcolspec#1{%
349 \ifx\tabucolX@spec\@empty\else \PackageWarning{tabu}
350 {Duplicate vertical alignment specification}\fi
351 \def\tabucolX@spec{#1}\expandafter\tabu@Xparse
352 }% \tabu@Xcolspec
353 \def\tabu@Xcoef#1{\def\@tempa{#1}%
354 \afterassignment\tabu@Xc@ef \tabu@cnt
355 }% \tabu@Xcoef
356 \def\tabu@Xc@ef{%
357 \advance\tabu@X@sum \@tempa\the\tabu@cnt\p@
358 \tabu@Xparse{}%
359 }% \tabu@Xc@ef
360 \def\tabucolX@do{%
361 \ifx\tabucolX@spec\@empty \def\tabucolX@spec{p}\fi
362 \ifdim \tabu@X@sum=\z@ \tabu@X@sum \p@\fi
363 \begingroup
364 \ifx \tabu@Xmath\relax
365 \edef\tabucolX@spec##1{\tabucolX@spec{##1}}%
366 \else
367 \edef\tabucolX@spec##1{>{\$}\tabucolX@spec{##1}<{\$}}%
368 \fi
369 \edef\tabu@temp{\dimexpr\strip@pt\tabu@X@sum \tabucolX \relax}%
370 \toks@ \expandafter{\tabucolX@align}%
371 \edef\x{%
372 \def\noexpand\tabu@temp{\the\toks@ \tabucolX@spec{\the\tabu@temp}}%
373 \def\noexpand\tabucolX@spec{\the\toks@ \tabucolX@spec{\tabu@temp}}%
374 }\expandafter\endgroup\x
375 }% \tabucolX@do

```

## 7.6 Before trials to reach the target

### Trial 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 \@halign` 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`.

```
376 \def\tabu@prep@TRIAL{%
```

```

377 \let\tabu@savedit\@preamble
378 \def\savetabu {\tabu@saveX}%
379 {\ifnum0='}\fi
380 \tabu@setup@TRIAL
381 \iftabu@spread % first trial only
382 \advance\tabu@X@cols \@ne
383 \expandafter\let\csname tabu@X@the\tabu@X@cols\endcsname \@undefined
384 \tabu@X@cols \z@ \tabu@naturalX \z@ \tabu@colX \z@
385 \let\tabu@naturalX@max \z@
386 \let\tabu@naturalX@min \z@
387 \let\tabu@startpboxORI \@startpbox
388 \let\tabu@startpbox \tabu@spread@startpbox
389 \else
390 \tabu@colX = \tabu@target
391 \fi
392 \tabu@cnt \z@ % number of trials
393 \def\tabu@lasttry{\m@ne\p}%
394 \tabu@arrayleft@measure
395 }% \tabu@prep@TRIAL

396 \def\tabu@setup@TRIAL{\tabu@global@X
397 \let\tabu@rewrite@X \tabu@rewrite@Xnested
398 \def\tabu@defaulttarget{\linewidth}%
399 \def\@elt##1{\global\value{##1}\the\value{##1}\relax}%
400 \xdef\tabu@global@X {\c@l@ckpt}\let\@elt \relax
401 \hbadness\@M \hfuzz\maxdimen
402 \let\hbadness \@tempcnta
403 \let\hfuzz \@tempdima
404 \let\write \tabu@nowrite
405 \let\@footnotetext\@gobble
406 \let\tabu@saveX \@gobble
407 \tabu@TRIAL@hook
408 }% \tabu@setup@TRIAL

```

\tabuDisableCommands

```

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

```

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

```

411 \def\tabu@arrayleft@measure#1\bgroup{%
412 \def\tabu@Xfinish{\ifnum0='}\fi}\tabu@global@X
413 \let\@halignto\tabu@halignto \tabu@firstclinettrue #1\bgroup}%
414 \toks@{\let\@preamble\tabu@savedit}% <required for multicolumn>
415 \tabu@collect
416 }% \tabu@arrayleft@measure

```

## Collecting the tabu body: required for X columns

\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.

```

417 \def\tabu@collect{\catcode'\^@=13\def\tabu@stack{b}\tabu@collectbody}
418 \long\def\tabu@collectbody#1\end#2{%
419 \edef\tabu@stack{\tabu@pushbegins #1\begin\end\expandafter\@gobble\tabu@stack}%
420 \ifx\tabu@stack\@empty
421 \toks@\expandafter{\the\toks@#1}\def\tabu@endenvir{\end{#2}}%
422 \expandafter\tabu@TRIAL
423 \else
424 \toks@\expandafter{\the\toks@#1\end{#2}}%

```

```

425     \expandafter\tabu@collectbody
426   \fi
427 }% \tabu@collectbody
428 \long\def\tabu@pushbegins#1\begin#2{\ifx\end#2\else b\expandafter\tabu@pushbegins\fi}

```

## 7.7 Measuring the tabu

### One trial after another

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

```

429 \def\tabu@TRIAL{\setbox\tabu@box \hbox\bgroup $\@arrayleft\vbox\bgroup \the\toks@
430                                     \tabu@endTRIAL}% constant
431 \def\tabu@endTRIAL{\endarray$\egroup
432   \iftabu@spread \tabu@spreadfalse
433     \let\@startpbox \tabu@startpboxORI
434     \tabu@spreadarith % <only once>
435   \else \tabu@arith
436   \fi
437   \iftabu@measuring \tabu@measuringfalse
438     \expandafter \tabu@TRIAL % <continue trials>
439   \else
440     \toks@\expandafter\expandafter\expandafter{%
441       \the\expandafter\toks@ \tabu@endenvir}%
442     \expandafter \tabu@Xfinish \the\toks@ % <we are then!>
443   \fi
444 }% \tabu@endTRIAL

```

### The arithmetic of X columns

`\tabu@arith` Algorithms.  
`\tabu@spreadarith`

```

445 \def\tabu@arith{%
446   \advance\tabu@cnt \@ne
447   \ifdim \tabu@lasttry=\wd\tabu@box
448     \tabu@message{\tabu@header Reached minimum width. Backing up}%
449   \else
450     \edef\tabu@lasttry{\the\wd\tabu@box}%
451     \tabu@message{\tabu@message@arith}%
452     \ifdim \dimexpr \wd \tabu@box - \tabu@target <\tabu@hfuzz
453       \tabu@message{\tabu@message@reached}%
454       \xdef\tabu@global@X {\tabu@target\the\tabu@target\relax \tabu@global@X
455                                     \tabucolX \the\tabucolX\relax}%
456     \else
457       \advance\tabucolX -\dimexpr(\wd\tabu@box-\tabu@target-\tabu@X@sum/65536)
458                                     *65536/\tabu@X@sum\relax
459       \ifdim \tabucolX<\z@
460         {\let\write\tabu@write\PackageWarning\tabu@
461           {X Columns too narrow (table too wide)\MessageBreak}}%
462         \tabucolX=1em%
463       \else \tabu@measuringtrue
464       \fi
465     \fi
466   \fi
467 }% \tabu@arith
468 \def\tabu@spreadarith{%

```





## Measuring the natural width for tabu spread

`\tabu@spread@startpbox` When “tabu spread” is used with  $X$  columns, the first trial must measure the natural width of the columns. Therefore, the `\vbox`, `\vtop` or `\vcenter` corresponding to the  $X$  columns definitions are temporarily changed into `\hbox`. More precisely, the natural width is measured with `\vbox{\hbox{...}}`.

For the further trials, the standard scheme for  $X$  column is used.

```

521 \def\tabu@spread@startpbox#1{\bgroup
522   \in@{\tabucolX}{#1}%
523   \ifin@
524     \let\vss\relax\let\vfil\vss\let\vfill\vss\let\vskip\@tempskipa
525     \hbox to\tabu@target\bgroup\aftergroup\tabu@spread@endhbox\hfil
526   \else
527     \expandafter\@gobble\tabu@startpboxORI{#1}% \@gobble \bgroup
528   \fi
529 }% \tabu@spread@startpbox
530 \def\tabu@spread@endhbox{%
531   \ifnum\badness>\@M %<text does not fit>
532     \global\tabu@naturalX \tabu@target
533   \else
534     \setbox\tabu@box\lastbox
535     \setbox\tabu@box\hbox{\unhbox\tabu@box}%
536     \global\advance\tabu@X@cols \@ne
537     \ifcsname tabu@X\the\tabu@X@cols\endcsname
538       \global\advance\tabu@naturalX \wd\tabu@box
539     \else
540       \global\tabu@X@cols \@ne
541       \global\tabu@naturalX \wd\tabu@box
542     \fi
543     \ifdim \tabu@naturalX@max<\tabu@naturalX
544       \xdef\tabu@naturalX@max{\the\tabu@naturalX}%
545     \fi
546     \ifdim \tabu@naturalX@min<\dimexpr \wd\tabu@box * \tabu@X@sum /
547       (65536*\csname tabu@X\the\tabu@X@cols\endcsname)\relax
548       \xdef\tabu@naturalX@min{\the\dimexpr \wd\tabu@box * \tabu@X@sum /
549         (65536*\csname tabu@X\the\tabu@X@cols\endcsname)\relax}%
550     \fi
551   \fi
552   \box\tabu@box
553   \egroup % end of \vtop (measure)
554 }% \tabu@spread@endhbox

```

## 7.8 Numbers in tabu

### `\tabudecimal`

#### `\tabudecimal`

```

555 \def\tabu@tabudecimal#1{%
556   \def\tabu@decimal{#1}\@temptokena{}%
557   \let\tabu@getdecimal@ \tabu@getdecimal@ignorespaces
558   \tabu@scandecimal
559 }% \tabudecimal
560 \def\tabu@scandecimal{\futurelet \tabu@temp \tabu@getdecimal@}
561 \def\tabu@skipdecimal#1{#1\tabu@scandecimal}
562 \def\tabu@getdecimal@ignorespaces{%
563   \ifcase 0\ifx\tabu@temp\ignorespaces\else
564     \ifx\tabu@temp\@sptoken1\else

```



```

565             2\fi\fi\relax
566         \let\tabu@getdecimal@ \tabu@getdecimal
567         \expandafter\tabu@skipdecimal
568     \or \expandafter\tabu@gobblespace\expandafter\tabu@scandecimal
569     \else \expandafter\tabu@skipdecimal
570     \fi
571 }% \tabu@getdecimal@ignorespaces
572 \def\tabu@get@decimal#1{\@temptokena\expandafter{\the\@temptokena #1}%
573             \tabu@scandecimal}
574 \def\do#1{%
575     \def\tabu@get@decimalspace#1{%
576         \@temptokena\expandafter{\the\@temptokena #1}\tabu@scandecimal}%
577 }\do{ }

```

\tabu@getdecimal

```

578 \def\tabu@getdecimal{%
579     \ifcase    0\ifx 0\tabu@temp\else
580             \ifx 1\tabu@temp\else
581             \ifx 2\tabu@temp\else
582             \ifx 3\tabu@temp\else
583             \ifx 4\tabu@temp\else
584             \ifx 5\tabu@temp\else
585             \ifx 6\tabu@temp\else
586             \ifx 7\tabu@temp\else
587             \ifx 8\tabu@temp\else
588             \ifx 9\tabu@temp\else
589             \ifx .\tabu@temp\else
590             \ifx ,\tabu@temp\else
591             \ifx -\tabu@temp\else
592             \ifx +\tabu@temp\else
593             \ifx e\tabu@temp\else
594             \ifx E\tabu@temp\else
595             \ifx\tabu@celleft\tabu@temp1\else
596             \ifx\ignorespaces\tabu@temp1\else
597             \ifx\@sptoken\tabu@temp2\else
598             3\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\relax
599         \expandafter\tabu@get@decimal
600     \or \expandafter\tabu@skipdecimal
601     \or \expandafter\tabu@get@decimalspace
602     \else\expandafter\tabu@printdecimal
603     \fi
604 }% \tabu@getdecimal
605 \def\tabu@printdecimal{%
606     \edef\tabu@temp{\the\@temptokena}%
607     \ifx\tabu@temp\@empty\else
608     \ifx\tabu@temp\space\else
609         \expandafter\tabu@decimal\expandafter{\the\@temptokena}%
610     \fi\fi
611 }% \tabu@printdecimal

```

## 7.9 Lines inside tabu

### Vertical lines

`\tabuvline@rewrite` | is defined as a new column type (only inside tabu) in order to add an optional argument: the width of the rule.

```
612 \newcommand*\tabuvline@rewrite[1][\]{\tabuvline@arg{#1}%
613 \expandafter \NC@find \tabu@temp}
614 \def\tabu@vline#1{\vrule width#1}
```

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

```
615 \def\tabuvline@arg#1{%
616 \ifx\#1\% <shortcut when no option>
617 \def\tabu@temp{\tabu@vline\arrayrulewidth}}%
618 \else \futurelet \tabu@temp \tabuvline@argi #1\p@\p@\@nnil \tabuvline@argiii
619 \fi
620 }% \tabuvline@arg
621 \def\tabuvline@argi{%
622 \let\tabu@color \@empty
623 \ifcat A\noexpand\tabu@temp
624 \@tempdima\arrayrulewidth
625 \expandafter\tabu@getlinecolor
626 \else \expandafter\tabuvline@argii
627 \fi
628 }% \tabuvline@argi
629 \def\tabuvline@argii#1\@nnil{%
630 \tabu@maybecolor \@tempdima #1 \@nnil
631 }% \tabuvline@argii
632 \def\tabuvline@argiii{%
633 \edef\tabu@temp{\color@begingroup\tabu@color
634 \noexpand\tabu@vline{\the\@tempdima}\color@endgroup}}%
635 }% \tabuvline@argiii
```

`\tabu@lines` This macro is used in `\tabu@setup` to “mount” the character | as a column type.

```
636 {
637 \expandafter\let\csname NC@find@|\endcsname\relax
638 \expandafter\let\csname NC@rewrite@|\endcsname\relax
639 \xdef\tabu@lines{%
640 \let\csname NC@find@|\endcsname \noexpand\tabuvlines@find
641 \let\csname NC@rewrite@|\endcsname \noexpand\tabuvline@rewrite
642 }% \tabu@vline
643 }
644 \def\tabuvlines@find#1|{\NC@{#1}}
```

### Horizontal lines: `\tabucline`

`\tabu@firstcline`

```
645 \def\tabu@firstcline{%
646 \tabu@firstclinettrue
647 \let\tabu@everycr\everycr
648 \def\tabu@restoreeverycr{\let\everycr\tabu@everycr}%
649 \def\everycr{\afterassignment\tabu@restoreeverycr\@temptokena}% <for ialign>
650 \tabu@everycr{\noalign{%
651 \global\tabu@everycr{\noalign{%
652 \global\tabu@everycr{}\global\tabu@firstclinefalse}}}%
653 \global\let\everycr\tabu@everycr}}%
654 }% \tabu@firstcline
```

`\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.

```

655 \def\tabu@linedash{4pt}%    <default value>
656 \def\tabu@dashgap{4pt}%    <default value>
657 \def\tabucline{\noalign{\ifnum0='}\fi\@ifstar
658   {\@tempswatrue\tabu@ccline}
659   {\@tempswafalse\tabu@ccline}}%
660 \newcommand*\tabu@ccline[2][]{%
661   \tabu@startstop{#2}\tabu@getlinespec{#1}\@multicnt\@ne
662   \ifnum\tabu@start>\tabu@stop \the\toks@
663   \else\ifx\tabu@xleaders\relax\the\toks@
664   \else
665     \toks@{\ifnum0='{ \fi}\noalign{\global\setbox\tabu@box \box\@arstrutbox}}%
666     \iftabu@firstccline\if\tabu@align t\vskip-\ht\@arstrutbox\fi\fi
667     \@whilenum\@multicnt<\tabu@start\do{\advance\@multicnt\@ne
668       \toks@\expandafter{\the\toks@ &}}%
669     \loop
670       \toks@\expandafter{\the\toks@ \omit }%
671       \ifcase 0\if@tempswa\else\ifnum \@multicnt>\tabu@start 1\fi\fi\relax
672       \expandafter \tabu@add \csname tabu@tok@\the\@multicnt L\endcsname \@nil
673       \fi
674       \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
675       \toks@\expandafter{\the\toks@ \color@begingroup}%
676       \toks@\expandafter\expandafter\expandafter{\expandafter\the
677         \expandafter\toks@ \tabu@xleaders \color@endgroup}%
678       \fi
679       \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi %
680       \expandafter \tabu@add \csname tabu@tok@\the\@multicnt R\endcsname \@nil
681       \fi
682       \ifnum\@multicnt<\tabu@stop
683         \advance\@multicnt\@ne
684         \if@tempswa \toks@\expandafter{\the\toks@ &}%
685         \else      \toks@\expandafter{\the\toks@ \span}\fi
686       \repeat
687       \@whilenum\@multicnt<\tabu@nbccls\do{\advance\@multicnt\@ne
688         \toks@\expandafter{\the\toks@ &}}%
689       \toks@\expandafter{\the\toks@ \cr \noalign{%
690         \global\tabu@firstcclinefalse \global\setbox\@arstrutbox \box\tabu@box}}%
691       \the\toks@
692     \fi\fi
693 }% \tabucline
694 \def\tabu@add #1\@nil{\toks@\expandafter{\the\toks@ #1\ifmmode$\fi}}
```

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

```

695 \def\tabu@startstop#1{\tabu@start@stop #1\relax 1-\tabu@nbccls\@nnil}
696 \def\tabu@start@stop #1-#2\@nnil{%
697   \countdef\tabu@start100\countdef\tabu@stop101 % <inside noalign group>
698   \@defaultunits \tabu@start \number0#1\relax\@nnil
699   \@defaultunits \tabu@stop \number0#2\relax\@nnil
700   \ifnum\tabu@start>\tabu@nbccls\tabu@start=\tabu@nbccls\else
701     \ifnum\tabu@start>\z@\else \tabu@start=\@ne \fi\fi
702   \ifnum\tabu@stop>\tabu@nbccls \tabu@stop\tabu@nbccls \else
703     \ifnum\tabu@stop>\z@\else \tabu@stop=\tabu@nbccls \fi\fi
704 }% \tabu@start@stop
```

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

```

705 \def\tabu@getlinespec#1{\let\tabu@xleaders \relax
706   \@defaultunits \let\@tempa=#1 \relax\@nnil
707   \ifx\@tempa\relax \let\tabu@xleaders \tabu@defaultleaders\else
708   \ifx\@tempa\hbox \tabu@defleaders{#1}\else
709   \ifx\@tempa\box \tabu@defleaders{#1}\else
710   \ifx\@tempa\copy \tabu@defleaders{#1}\else
711   \ifcsname tabu@line@style@\string#1\endcsname
712     \csname tabu@line@style@\string#1\endcsname
713   \else \tabu@getline{#1}\p@ on0pt off0pt}\fi\fi\fi\fi\fi
714 }% \tabu@getlinespec
715 \def\tabu@defleaders#1{%
716   \def\tabu@xleaders{\xleaders\hbox{\lower.5\extrarowheight#1}\tabu@leaderfill}}
717 \def\tabucline@warn#1{\PackageWarning{tabu}
718   {Undefined line syle: #1
719   \MessageBreak Using default line style instead}}%
720 \let\tabu@xleaders \tabu@defaultleaders
721 }% \tabucline@warn

```

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

```

722 \def\tabu@getline#1{\tabu@lineon #1 \@nil on\tabu@linedash \p@ \@nil\@nnil{#1}}
723 \def\tabu@maybecolor{\ifx\tabu@color\@empty \afterassignment\tabu@colortest
724   \else \@defaultunits\fi}
725 \def\tabu@lineon #1on#2\@nil#3\@nnil#4{%
726   \let\tabu@color \@empty
727   \@defaultunits \let\@tempa=#1 \relax\@nnil
728   \ifcase 0\ifx o\@tempa 1\else\ifcat A\noexpand\@tempa 2\fi\fi\relax
729     \tabu@maybecolor \@tempdima #1\arrayrulewidth \p@\@nnil
730     \tabu@maybecolor \@tempdimb #2\p@ \@nnil
731     \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil
732   \or\@tempdima \arrayrulewidth
733     \tabu@maybecolor \@tempdimb #2\p@ \@nnil
734     \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil
735   \else
736     \tabu@maybecolor \@tempdima \arrayrulewidth #1\p@\@nnil
737     \ifx\tabu@color\@empty \tabucline@warn{#4}\else
738       \@tempdimb \z@
739     \tabu@lineoff off0pt \p@\@nnil \fi
740   \fi
741 }% \tabu@lineon
742 \def\tabu@lineoff #1off#2\@nnil{%
743   \tabu@maybecolor \@tempdimc #2\p@ \@nnil
744   \ifdim \@tempdimb=\z@
745     \ifdim \@tempdimc>\z@ \@tempdimb \tabu@dashgap\relax\fi\fi
746   \ifdim \@tempdimc=\z@
747     \ifdim \@tempdimb>\z@ \@tempdimc \tabu@linedash\relax\fi\fi
748   \ifdim \@tempdima<\z@ \else
749   \ifdim \@tempdimb<\z@ \else
750   \ifdim \@tempdimc<\z@ \else
751     \edef \tabu@xleaders{\tabu@color\xleaders
752       \ifdim\@tempdimc>\z@
753         \hbox\bgroup \kern\the\dimexpr\@tempdimc/2\relax\fi
754       \noexpand\iftabu@firstcline
755       \vrule depth\the\@tempdima
756       \ifdim\@tempdimb>\z@ width\the\@tempdimb\fi

```

```

757         \noexpand\else
758         \vrule height\dimexpr-\extrarowheight+\the\@tempdima
759             depth\extrarowheight
760             \ifdim\@tempdimb>\z@ width\the\@tempdimb\fi
761         \noexpand\fi
762         \ifdim\@tempdimc>\z@
763             \kern\the\dimexpr\@tempdimc/2\egroup\fi
764         \tabu@leaderfill}%
765     \fi\fi\fi
766 }% \tabu@lineoff

```

`\tabu@colortest`

```

767 \def\tabu@colortest{\futurelet\tabu@temp\tabu@linecolor}
768 \def\tabu@linecolor{%
769     \ifcase 0\if ,\noexpand\tabu@temp\else
770         \ifx\relax\tabu@temp\else
771         \ifx \@sptoken\tabu@temp1\else
772         \ifcat A\noexpand\tabu@temp2\else
773             3\fi\fi\fi\fi\relax
774     \def\tabu@next##1{\futurelet\tabu@temp\tabu@linecolor}%
775     \or\def\tabu@next{\tabu@gobblespace{\futurelet\tabu@temp\tabu@linecolor}}%
776     \or\let\tabu@next\tabu@getlinecolor
777     \else\expandafter\remove@to@nnil
778     \fi \tabu@next
779 }% \tabu@linecolor
780 \def\tabu@getlinecolor#1p@{%
781     \edef\@tempa{\zap@space #1 \@empty}%
782     \ifcsname\string\color@\@tempa\endcsname
783         \edef\tabu@color{\noexpand\noexpand\noexpand\color{\@tempa}}% \set@color
784     \fi\remove@to@nnil
785 }% \tabu@getlinecolor

```

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

```

786 \def\tabulinestyle#1{\@for\@tempa:=#1\do{\expandafter\tabu@linestyle\@tempa==\@nil}}
787 \def\tabu@linestyle#1=#2=#3\@nil{%
788     \begingroup \tabu@getlinespec {#2}\expandafter\gdef
789         \csname tabu@line@style@\string#1\expandafter\endcsname
790             \expandafter{\expandafter\def\expandafter\tabu@xleaders
791                 \expandafter{\tabu@xleaders}}%
792     \endgroup
793 }% \tabu@linestyle
794 \expandafter\def \csname tabu@line@style@\endcsname {%
795     \let\tabu@xleaders \tabu@defaultleaders}%
796 \def\tabu@defaultleaders{\leaders
797     \iftabu@firstcline
798         \vrule depth \arrayrulewidth
799     \else
800         \vrule height\dimexpr-\extrarowheight+\arrayrulewidth
801         depth \extrarowheight
802     \fi
803     \tabu@leaderfill}
804 \let\tabu@leaderfill \hfil

```

## 7.10 Verbatim inside tabu with X columns

`\tabu@sanitizetext`

```

805 {\catcode32=13\relax\catcode'\^@=13\relax
806 \gdef\tabu@verb{\@sanitize\makeatletter\catcode'\^=7\edef\^{\string^}%
807 \catcode32=13\let =\ \catcode'\^@=13\def^^@{\par}\endlinechar\m@ne}%
808 }
809 \newcommand\tabu@sanitizetext[1][\ttfamily]{\begingroup
810 \tabu@verb #1\tabu@s@nitizetext}
811 \long\def\tabu@s@nitizetext#1{\@makeother\{\@makeother\}%
812 \everyeof{\noexpand}\scantokens{#1}\endgroup}

```

## 7.11 \savetabu

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

`\tabu@saveX` `\tabu@saveX` replaces `\savetabu` inside a tabu with X columns. The X columns widths have to be stored, that is, the value in points of `\tabucolX`.

```

813 \newcommand*\savetabu[1]{\noalign{%
814 \ifx\#1\\ \tabu@savewarn{\The tabu will not be saved}\else
815 \ifundefined\tabu@save@\string#1\}{\tabu@savewarn{#1}{Overwritting}}%
816 {\toks@\expandafter{\tabu@sav@}}%
817 \expandafter\xdef\csname tabu@sav@@\string#1\endcsname{%
818 \tabu@target\the\tabu@target\relax
819 \the\toks@}}%
820 \fi}%
821 }% \savetabu

822 \def\tabu@saveX#1{\noalign{%
823 \ifx\#1\\ \tabu@savewarn{\The tabu will not be saved}\else
824 \ifundefined\tabu@sav@@\string#1\}{\tabu@savewarn{#1}{Overwritting}}%
825 {\toks@\expandafter{\tabu@sav@}}%
826 \expandafter\xdef\csname tabu@sav@@\string#1\endcsname{%
827 \tabucolX\the\tabucolX\relax
828 \tabu@target\the\tabu@target\relax
829 \the\toks@}}%
830 \fi}%
831 }% \tabu@saveX

```

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

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

```

832 \def\tabu@savewarn#1#2{\PackageInfo{tabu}
833 {User-name '#1' already used for \string\savetabu
834 \MessageBreak #2}}%
835 \def\tabu@saveerr#1{\PackageError{tabu}
836 {User-name '#1' is unknown for \string\usetabu
837 \MessageBreak I cannot restore an unknown preamble!}\@ehd}

```

## 7.12 \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`.

```

838 \def\tabu@rowfont{\noalign{\ifnum0='}\fi\tabu@row@font}
839 \newcommand*\tabu@row@font[2][]{%
840   \global\tabu@everycr@tok=\everycr
841   \global\let\tabu@celleft \tabu@celleft
842   \global\let\tabu@celright \tabu@celright
843   \gdef\tabu@cellfont{#2}%
844   \ifcsname tabu@cell@#1\endcsname % row alignment
845     \csname tabu@cell@#1\endcsname \fi
846   \toks@\expandafter {\tabu@celleft\tabu@cellfont}% inside \noalign group ok
847   \xdef\tabu@celleft {\the\toks@}%
848   \global\everycr\expandafter {\the\everycr\tabu@rowfont@reset}%
849   \ifnum0='{ \fi}% end of noalign group
850 }% \rowfont

```

`\tabu@rowfont@reset` This macro resets `\tabu@celllalign`, `\tabu@celleft`, `\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.

```

851 \def\tabu@rowfont@reset{%
852   \noalign{%
853     \global\let\tabu@celleft \tabu@celleft
854     \global\let\tabu@celright \tabu@celright
855     \global\let\tabu@cellfont \@empty
856     \global\let\tabu@celllalign \@empty
857     \global\let\tabu@cellralign \@empty
858     \global\everycr=\tabu@everycr@tok
859     \global\tabu@everycr@tok{}%
860   }%
861 }% \tabu@rowfont@reset

```

### 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@celleft`, `\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.

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

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.

```

862 \newif\iftabu@cellright
863 \AtBeginDocument{\let\array@prepnext@tok \prepnext@tok}% original definition
864 \def\tabu@prepnext@tok{%
865   \ifnum \count@<\z@ % <first initialisation>
866     \@tempcnta \@M % <not initialized by array.sty>
867     \tabu@nbc\z@
868     \gdef\tabu@global@temp{\tabu@nbc\z@}%
869     \expandafter\let\csname tabu@tok@1\endcsname \relax
870     \tabu@cellrightfalse
871   \else
872     \ifcase \numexpr \count@-\@tempcnta \relax % (case 0): prev. token is left
873       \advance \tabu@nbc\@ne
874       \ifnum \tabu@nbc>\@ne
875         \expandafter\let\csname tabu@tok@\the\tabu@nbc L\endcsname \relax
876       \fi
877       \expandafter\let\csname tabu@tok@\the\tabu@nbc R\endcsname \relax
878       \expandafter\gdef\expandafter\tabu@global@temp\expandafter{%
879         \tabu@global@temp \advance\tabu@nbc\@ne}%
880       \iftabu@cellright % before-previous token is right and is finished
881         \tabu@cellrightfalse % <only once>
882         \tabu@savetok R\tabu@preptokenright
883       \fi
884       \tabu@savetok L\tabu@preptokenleft
885     \or % (case 1) previous token is right
886       \tabu@savetok R\tabu@cellrighttrue
887     \else % special column: do not change the token
888       \ifnum \tabu@nbc>\z@ %special column: always on the right of normal one
889       \else % unless this is the very first column (\tabu@nbc=0)
890         \advance\tabu@nbc\@ne \tabu@savetok L\advance\tabu@nbc\m@ne
891       \fi
892       \iftabu@cellright % before-previous token is right
893         \tabu@cellrightfalse
894       \tabu@savetok R\tabu@preptokenright
895     \fi

```

```

896     \fi % \ifcase
897     \fi
898     \array@prepnext@tok
899 }% \tabu@prepnext@tok
900 \def\tabu@preptokenright{%
901     \advance \count@ \m@ne
902     \toks\count@\expandafter {\the\toks\count@ \tabu@cellright \tabu@celllralign}%
903     \advance \count@ \@ne
904 }% \tabu@preptokenright
905 \def\tabu@preptokenleft{\toks\count@\expandafter {\expandafter\tabu@celllalign
906                                     \the\toks\count@ \tabu@cellleft}%
907 }% \tabu@preptokenleft
908 \def\tabu@savetok#1{\begingroup
909     \expandafter\tabu@savetok\csname tabu@tok@\the\tabu@nbc\cols #1\endcsname
910 }% \tabu@savetok
911 \def\tabu@savetok#1{%
912     \@temptokena\toks\count@
913     \ifx#1\relax\else \@temptokena\expandafter\expandafter
914         \expandafter{\expandafter#1\the\@temptokena}%
915     \fi
916     \@temptokena\expandafter{\expandafter\def\expandafter#1\expandafter{%
917                                     \the\@temptokena}}%
918     \toks@\expandafter\expandafter\expandafter{\expandafter\tabu@global@temp
919                                     \the\@temptokena}%
920     \xdef\tabu@global@temp{\the\toks@}%
921     \expandafter\endgroup \the\@temptokena
922 }% \tabu@savetok

```

## Neutralisation of glues and alignment modification

<code>\tabu@cellleft</code> <code>\tabu@celllalign</code> <code>\tabu@cellright</code> <code>\tabu@celllralign</code> <code>\tabu@cellfont</code>	<p>First initialisation to <code>\@empty</code>.</p> <pre> 923 \let\tabu@cellleft\@empty 924 \let\tabu@cellright\@empty 925 \def\tabu@celllalign{\tabu@cellleft}% row font spec. applies to pre-column material 926 \let\tabu@celllralign\@empty 927 \let\tabu@cellfont\@empty </pre>
<code>\tabu@cell@l</code> <code>\tabu@cell@c</code> <code>\tabu@cell@r</code> <code>\tabu@cell@j</code>	<p>Setup macros to modify the alignment. The skips inserted to make the standard alignment specified in the tabular preamble are not the same with standard array tabulars and colortbl tabulars, hence the switch <code>\iftabu@colortbl</code>.</p> <pre> 928 \def\tabu@cell@l{% force alignment to left 929     \gdef\tabu@celllalign{\tabu@removehfil 930         \raggedright\arraybackslash 931         \tabu@cellleft}% 932     \gdef\tabu@celllralign{\tabu@flush1\tabu@ignorehfil}% 933     \toks@\expandafter{\tabu@cellleft\raggedright\arraybackslash}% local 934     \xdef\tabu@cellleft{\the\toks@}% 935 }% \tabu@cell@l 936 \def\tabu@cell@c{% force alignment to center 937     \gdef\tabu@celllalign{\tabu@removehfil 938         \centering\arraybackslash 939         \tabu@flush{.5}\tabu@cellleft}% 940     \gdef\tabu@celllralign{\tabu@flush{.5}\tabu@ignorehfil}% 941     \toks@\expandafter{\tabu@cellleft\centering\arraybackslash}% local 942     \xdef\tabu@cellleft{\the\toks@}% 943 }% \tabu@cell@c 944 \def\tabu@cell@r{% force alignment to right </pre>

```

945 \gdef\tabu@celllalign{\tabu@removehfil
946 \raggedleft\arraybackslash
947 \tabu@flush1\tabu@cellleft}%
948 \gdef\tabu@celllralign{\tabu@ignorehfil}%
949 \toks@\expandafter{\tabu@cellleft\raggedleft\arraybackslash}% local
950 \xdef\tabu@cellleft{\the\toks@}%
951 }% \tabu@cell@r
952 \def\tabu@cell@j{% force justification (for p, m, b columns)
953 \gdef\tabu@celllalign{\tabu@justify\tabu@cellleft}%
954 \global\let\tabu@celllralign\@empty
955 \toks@\expandafter{\tabu@cellleft\tabu@justify}% local (noalign grp)
956 \xdef\tabu@cellleft{\the\toks@}%
957 }% \tabu@cell@j
958 \def\tabu@justify{%
959 \leftskip\z@skip \@rightskip\leftskip \rightskip\@rightskip
960 \parfillskip\@flushglue
961 }% \tabu@justify
962 %% ragged2e settings
963 \def\tabu@cell@L{% force alignment to left (ragged2e)
964 \gdef\tabu@celllalign{\tabu@removehfil
965 \RaggedRight\arraybackslash
966 \tabu@cellleft}%
967 \gdef\tabu@celllralign{\tabu@flush1\tabu@ignorehfil}%
968 \toks@\expandafter{\tabu@cellleft\RaggedRight\arraybackslash}%
969 \xdef\tabu@cellleft{\the\toks@}%
970 }% \tabu@cell@L
971 \def\tabu@cell@C{% force alignment to center (ragged2e)
972 \gdef\tabu@celllalign{\tabu@removehfil
973 \Centering\arraybackslash
974 \tabu@flush{.5}\tabu@cellleft}%
975 \gdef\tabu@celllralign{\tabu@flush{.5}\tabu@ignorehfil}%
976 \toks@\expandafter{\tabu@cellleft\Centering\arraybackslash}%
977 \xdef\tabu@cellleft{\the\toks@}%
978 }% \tabu@cell@C
979 \def\tabu@cell@R{% force alignment to right (ragged2e)
980 \gdef\tabu@celllalign{\tabu@removehfil
981 \RaggedLeft\arraybackslash
982 \tabu@flush1\tabu@cellleft}%
983 \gdef\tabu@celllralign{\tabu@ignorehfil}%
984 \toks@\expandafter{\tabu@cellleft\RaggedLeft\arraybackslash}%
985 \xdef\tabu@cellleft{\the\toks@}%
986 }% \tabu@cell@R
987 \def\tabu@cell@J{% force justification (ragged2e)
988 \gdef\tabu@celllalign{\justifying\arraybackslash\tabu@cellleft}%
989 \global\let\tabu@celllralign\@empty
990 \toks@\expandafter{\tabu@cellleft\justifying\arraybackslash}%
991 \xdef\tabu@cellleft{\the\toks@}%
992 }% \tabu@cell@J
993 \def\tabu@flush#1{%
994 \iftabu@colortbl % colortbl uses \hfill rather than \hfil
995 \hskip \ifnum\currentgrouptype>13 \stretch{#1}%
996 \else\ifdim#1pt<1pt \tabu@cellskip
997 \else \stretch{#1}
998 \fi\fi \relax
999 \else % array.sty
1000 \ifnum \currentgrouptype>13\relax
1001 \hfil \hskip1sp
1002 \fi

```

```

1003   \fi
1004 }% \tabu@flush
1005 \AtBeginDocument{%
1006   \@ifpackageloaded{ragged2e}
1007     {}
1008     {\let\tabu@cell@L \tabu@cell@l
1009       \let\tabu@cell@R \tabu@cell@r
1010       \let\tabu@cell@C \tabu@cell@c
1011       \let\tabu@cell@J \tabu@cell@j
1012     }%
1013 }% 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.

```

1014 \newskip\tabu@cellskip
1015 \let\tabu@hfil\hfil
1016 \let\tabu@hfill\hfill
1017 \let\tabu@hskip\hskip
1018 \def\tabu@removehfil{%
1019   \iftabu@colortbl
1020     \unkern \tabu@cellskip = \lastskip
1021     \ifnum\gluestretchorder\tabu@cellskip = \tw@ \hskip-\tabu@cellskip
1022     \else \tabu@cellskip = \z@skip
1023   \fi
1024   \else
1025     \ifdim\lastskip=1sp\unskip\fi
1026     \ifnum\gluestretchorder\lastskip = \@ne
1027       \hfilneg % \hfilneg for array.sty but not for colortbl...
1028     \fi
1029   \fi
1030 }% \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.

```

1031 \def\tabu@ignorehfil{%
1032   \aftergroup\tabu@nohfil
1033 }% \tabu@ignorehfil
1034 \def\tabu@nohfil{% \hfil -> do nothing + restore original \hfil
1035   \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1036 }% \tabu@nohfil
1037 \AtBeginDocument{%
1038   \@ifpackageloaded{colortbl}
1039     {%
1040       \def\tabu@nohfil{%
1041         \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1042         \def\hfill{\let\hfill\tabu@hfill}% (colortbl uses \hfill) pfff...
1043         \def\hskip##1\relax{\let\hskip\tabu@hskip}}% local
1044       }% \@ifpackageloaded colortbl
1045     }%
1046 }% AtBeginDocument

```

## 7.13 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.

```

1047 \def\tabu@fbox{%
1048   \leavevmode
1049   \let\color@bgroup\bgroup
1050   \def\color@egroup{\endgraf\egroup}%
1051   \afterassignment\tabu@begin@fbox
1052   \setbox\@tempboxa \hbox
1053 }% \tabu@fbox
1054 \def\tabu@begin@fbox{\color@bgroup\kern\fboxsep\aftergroup\tabu@end@fbox}
1055 \def\tabu@end@fbox{\kern\fboxsep\color@egroup\@framebex\relax}

```

### \centering, \raggedright, \raggedleft

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

```

1056 \expandafter\def\expandafter\tabu@centering\expandafter{%
1057   \centering\arraybackslash}
1058 \expandafter\def\expandafter\tabu@raggedleft\expandafter{%
1059   \raggedleft\arraybackslash}
1060 \expandafter\def\expandafter\tabu@raggedright\expandafter{%
1061   \raggedright\arraybackslash}
1062 \AtBeginDocument{%
1063 \expandafter\def\expandafter\tabu@trivlist\expandafter{%
1064   \expandafter\let\expandafter\\\expandafter\@centercr\@trivlist}%
1065 }%

```

## 7.14 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.

```

1066 \def\tabu@fix@arrayright{%
1067   \ifpackageloaded{arydshln}
1068   {%
1069     \ifpackageloaded{colortbl}
1070     {% colortbl + arydshln
1071       \def\tabu@endarray{%
1072         \adl@endarray \egroup \adl@arrayrestore \CT@end \egroup %<original>
1073         \@arrayright      % <FC>
1074         \gdef\@preamble{}}% <FC>
1075       }% \endarray
1076     {% arydshln / no colortbl
1077       \def\tabu@endarray{%
1078         \adl@endarray \egroup \adl@arrayrestore \egroup %<original>
1079         \@arrayright      % <FC>
1080         \gdef\@preamble{}}% <FC>
1081       }% \endarray
1082   }%

```

```

1083   {%
1084   \@ifpackageloaded{colortbl}
1085     {% colortbl / no arydshln
1086     \def\tabu@endarray{%
1087       \crrc \egroup \egroup
1088       \@arrayright % <FC>
1089       \gdef\@preamble{}\CT@end
1090     }}%
1091   {\PackageWarning{tabu}
1092     {\string\@arrayright\space is missing from the
1093     \MessageBreak definition of \string@endarray.
1094     \MessageBreak Comptability with delarray.sty is broken.}}%
1095   }%
1096 }% \tabu@fix@arrayright

```

## arydshln @ columns

```

1097 \def\tabu@adl@xarraydashrule#1#2#3{%
1098   \ifnum\@lastchclass=\adl@class@start\else
1099   \ifnum\@lastchclass=\@ne\else
1100   \ifnum\@lastchclass=5 \else % <FC> @-arg (class 5) and !-arg (class 1)
1101     \adl@leftrulefalse \fi \fi % must be treated the same
1102   \fi
1103   \ifadl@zwvrule\else \ifadl@inactive\else
1104     \@addtopreamble{\vrule\@width\arrayrulewidth
1105       \@height\z@ \@depth\z@}\fi \fi
1106   \ifadl@leftrule
1107     \@addtopreamble{\adl@vlineL{\CT@arc@}{\adl@dashgapcolor}%
1108       {\number#1}#3}%
1109   \else \@addtopreamble{\adl@vlineR{\CT@arc@}{\adl@dashgapcolor}%
1110     {\number#2}#3}
1111   \fi
1112 }% \tabu@adl@xarraydashrule

```

## arydshln 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).

```

1113 \def\tabu@adl@act@endpbox{%
1114   \unskip \ifhmode \nobreak \fi \@finalstrut \@arstrutbox
1115   \egroup
1116   \adl@colhtdp \box\adl@box \hfil
1117 }% \tabu@adl@act@endpbox

```

1118 </package>

## 8 History

[2010/10/28 v1.0]

- First version.

## 9 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>\@arrayleft</code> . . . . .	429
<code>\@arrayright</code> . . . . .	46, 1073, 1079, 1088, 1092
<code>\@arstrutbox</code> 75, 78, 137, 142, 665, 666, 690, 1114	
<code>\@centercr</code> . . . . .	179, 1064
<code>\@defaultunits</code> . . . . .	698, 699, 706, 724, 727
<code>\@elt</code> . . . . .	399, 400
<code>\@finalstrut</code> . . . . .	1114
<code>\@flushglue</code> . . . . .	120, 121, 122, 960
<code>\@footnotetext</code> . . . . .	170, 405
<code>\@halignto</code> . . . . .	115, 117, 213, 214, 217, 218, 267, 290, 413
<code>\@let@token</code> . . . . .	198, 200, 201, 202, 203, 204, 307, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324
<code>\@m</code> . . . . .	509
<code>\@makeoother</code> . . . . .	811
<code>\@mpfn</code> . . . . .	239
<code>\@multicnt</code> . . . . .	661, 667, 671, 672, 674, 679, 680, 682, 683, 687
<code>\@preamble</code> . . . . .	377, 414, 1074, 1080, 1089
<code>\@rightskip</code> . . . . .	959
<code>\@sanitize</code> . . . . .	806
<code>\@sptoken</code> . . . . .	202, 320, 564, 597, 771
<code>\@tempcnta</code> . . . . .	402, 866, 872
<code>\@tempdimb</code> . . . . .	730, 733, 738, 744, 745, 747, 749, 756, 760
<code>\@tempdimc</code> . . . . .	743, 745, 746, 747, 750, 752, 753, 762, 763
<code>\@tempskipa</code> . . . . .	524
<code>\@temptokena</code> . . . . .	556, 572, 576, 606, 609, 649, 912, 913, 914, 916, 917, 919, 921
<code>\@trivlist</code> . . . . .	179, 1064
<code>\@undefined</code> . . . . .	5, 383
<code>\@whilenum</code> . . . . .	139, 667, 687
<code>\@xfootnotenext</code> . . . . .	171
<code>\{</code> . . . . .	811
<code>\}</code> . . . . .	811
<code>\^</code> . . . . .	417, 805, 806, 807
<code>\_</code> . . . . .	807
<code>\adl@box</code> . . . . .	1116
<code>\adl@class@start</code> . . . . .	1098
<code>\adl@colhtdp</code> . . . . .	1116
<code>\adl@dashgapcolor</code> . . . . .	1107, 1109
<code>\adl@endarray</code> . . . . .	1072, 1078
<code>\adl@leftrulefalse</code> . . . . .	1101
<code>\adl@vlineL</code> . . . . .	1107
<code>\adl@vlineR</code> . . . . .	1109
<code>\adl@xarraydashrule</code> . . . . .	151
<code>\afterassignment</code> 131, 132, 215, 354, 649, 723, 1051	
<code>\aftergroup</code> . . . . .	81, 98, 99, 174, 251, 268, 291, 525, 1032, 1054
<code>\array</code> . . . . .	232
<code>\array@prepnext@tok</code> . . . . .	863, 898
<code>\arraybackslash</code> . . . . .	930, 933, 938, 941, 946, 949, 965, 968, 973, 976, 981, 984, 988, 990, 1057, 1059, 1061
<code>\arraycolsep</code> . . . . .	224
<code>\arrayrulewidth</code> . . . . .	74, 226, 617, 624, 729, 732, 736, 798, 800, 1104
<code>\AtEndOfPackage</code> . . . . .	5, 23, 27, 30

### B

<code>\badness</code> . . . . .	531
<code>\begin</code> . . . . .	3

### C

<code>\catcode</code> . . . . .	10, 12, 417, 805, 806, 807
<code>\Centering</code> . . . . .	336, 973, 976
<code>\centering</code> . . . . .	176, 331, 938, 941, 1057
<code>\cl@ckpt</code> . . . . .	400
<code>\color</code> . . . . .	783
<code>\color@</code> . . . . .	782
<code>\color@begingroup</code> . . . . .	633, 675
<code>\color@bgroup</code> . . . . .	1049, 1054
<code>\color@egroup</code> . . . . .	1050, 1055
<code>\color@endgroup</code> . . . . .	634, 677
<code>\copy</code> . . . . .	710
<code>\countdef</code> . . . . .	697
<code>\crr</code> . . . . .	1087
<code>\currentgrouptype</code> . . . . .	995, 1000

### A

<code>\adl@act@@endpbox</code> . . . . .	153
<code>\adl@act@endpbox</code> . . . . .	152
<code>\adl@arrayrestore</code> . . . . .	1072, 1078

### D

<code>\d@llarend</code> . . . . .	190
<code>\delarray_(package_option)</code> . . . . .	26
<code>\doublerulesep</code> . . . . .	80, 229



<b>E</b>	
<code>\endarray</code>	46, 47, 146, 154, 431, 1075, 1081, 1093
<code>\endlinechar</code>	807
<code>\endtabu</code>	145
<code>\endtabular</code>	147
<code>\everycr</code>	136, 647, 648, 649, 653, 840, 848, 858
<code>\everyeof</code>	812
<code>\everypar</code>	162, 163
<code>\extracolsep</code>	120, 121, 122
<code>\extrarowheight</code>	75, 76, 77, 227, 716, 758, 759, 800, 801
<code>\extratabsurround</code>	79, 228
<b>F</b>	
<code>\footnotetext</code>	239, 243
<code>\futurelet</code>	110, 198, 307, 560, 618, 767, 774, 775
<b>G</b>	
<code>\gluestretchorder</code>	1021, 1026
<b>H</b>	
<code>\hbadness</code>	401, 402
<code>\hfilneg</code>	1027
<code>\hfuzz</code>	186, 401, 403
<code>\ht</code>	75, 78, 666
<b>I</b>	
<code>\ifadl@inactive</code>	1103
<code>\ifadl@leftrule</code>	1106
<code>\ifadl@zwvrule</code>	1103
<code>\ifcase</code>	200, 310, 563, 579, 671, 674, 679, 728, 769, 872, 896
<code>\ifcat</code>	190, 324, 623, 728, 772
<code>\ifcsname</code>	537, 711, 782, 844
<code>\ifdim</code>	118, 161, 165, 186, 217, 261, 362, 447, 452, 459, 472, 474, 478, 482, 543, 546, 744, 745, 746, 747, 748, 749, 750, 752, 756, 760, 762, 996, 1025
<code>\ifin@</code>	47, 523
<code>\ifnum</code>	379, 412, 507, 508, 509, 512, 515, 531, 657, 662, 665, 671, 674, 679, 682, 700, 701, 702, 703, 838, 849, 865, 874, 888, 995, 1000, 1021, 1026, 1098, 1099, 1100
<code>\iftabu@cellright</code>	862, 880, 892
<code>\iftabu@colortbl</code>	127, 994, 1019
<code>\iftabu@firstcline</code>	59, 666, 754, 797
<code>\iftabu@measuring</code>	59, 251, 256, 272, 437
<code>\iftabu@nested</code>	59, 172
<code>\iftabu@spread</code>	59, 245, 381, 432
<code>\ignorespaces</code>	563, 596
<code>\in@</code>	46, 522
<b>J</b>	
<code>\justifying</code>	988, 990
<b>L</b>	
<code>\lastbox</code>	164, 534
<code>\lastskip</code>	1020, 1025, 1026
<code>\leaders</code>	796
<code>\leftskip</code>	959
<code>\light_(package_option)</code>	29
<code>\linegoal</code>	24
<code>\linegoal_(package_option)</code>	22
<code>\linewidth</code>	20, 24, 158, 161, 165, 398
<code>\loop</code>	669
<code>\lower</code>	716
<b>M</b>	
<code>\maxdimen</code>	401
<code>\message</code>	19
<b>N</b>	
<code>\NC@find</code>	66, 100, 125, 231, 613
<code>\NC@rewrite@X</code>	97
<code>\newcolumntype</code>	45
<code>\newcount</code>	51, 52, 53
<code>\newdimen</code>	54, 55, 56, 57, 58
<code>\newsavebox</code>	63
<code>\newskip</code>	1014
<code>\newtoks</code>	127, 236
<code>\noalign</code>	135, 142, 650, 651, 657, 665, 689, 813, 822, 838, 846, 852
<code>\numexpr</code>	872
<b>O</b>	
<code>\omit</code>	670
<b>P</b>	
<code>\PackageError</code>	39, 41, 835
<code>\PackageInfo</code>	832
<code>\PackageWarning</code>	344, 349, 460, 717, 1091
<code>\parfillskip</code>	960
<code>\prepnext@tok</code>	31, 183, 863
<code>\ProcessOptions</code>	43
<b>R</b>	
<code>\RaggedLeft</code>	337, 981, 984
<code>\raggedleft</code>	178, 332, 946, 949, 1059
<code>\RaggedRight</code>	335, 965, 968
<code>\raggedright</code>	177, 330, 930, 933, 1061
<code>\remove@to@nnil</code>	326, 777, 784
<code>\repeat</code>	686
<code>\rightskip</code>	959
<code>\rowfont</code>	10, 40, 184, 838
<b>S</b>	
<code>\save@decl</code>	32, 96
<code>\savetabu</code>	9, 378, 813, 833
<code>\scantokens</code>	812
<code>\set@color</code>	783
<code>\setbox</code>	137, 142, 164, 429, 534, 535, 665, 690, 1052
<code>\setlength</code>	262, 282
<code>\span</code>	685
<code>\stretch</code>	995, 997
<code>\strip@pt</code>	278, 289, 369
<b>T</b>	
<code>\tabcolsep</code>	225
<code>\tabu</code>	145
<code>\tabu@@begin</code>	219, 221, 235
<code>\tabu@@cellleft</code>	841, 853

<code>\tabu@@cellright</code>	842, 854	<code>\tabu@flush</code>	932,
<code>\tabu@add</code>	672, 680, 694		939, 940, 947, 967, 974, 975, 982, 993, 1004
<code>\tabu@adl@act@endpbox</code>	152, 153, 1113, 1117	<code>\tabu@footnotes</code>	173, 174, 236
<code>\tabu@adl@xarraydashrule</code>	151, 1097, 1112	<code>\tabu@footnotetext</code>	170, 237
<code>\tabu@align</code>	222, 666	<code>\tabu@get@decimal</code>	572, 599
<code>\tabu@aligndefault</code>	158, 160, 219	<code>\tabu@get@decimalspace</code>	575, 601
<code>\tabu@arith</code>	435, 445	<code>\tabu@getdecimal</code>	566, 578
<code>\tabu@arrayleft@measure</code>	394, 411	<code>\tabu@getdecimal@</code>	557, 560, 566
<code>\tabu@begin</code>	198	<code>\tabu@getdecimal@ignorespaces</code>	557, 562, 571
<code>\tabu@begin@fbox</code>	1051, 1054	<code>\tabu@getline</code>	713, 722
<code>\tabu@box</code>	63, 137, 142, 429, 447, 450, 452, 457, 472, 473, 482, 492, 495, 499, 534, 535, 538, 541, 546, 548, 552, 665, 690	<code>\tabu@getlinecolor</code>	625, 776, 780, 785
<code>\tabu@cell@C</code>	971, 978, 1010	<code>\tabu@getlinespec</code>	661, 705, 788
<code>\tabu@cell@c</code>	928	<code>\tabu@gettarget</code>	213, 214, 215
<code>\tabu@cell@J</code>	987, 992, 1011	<code>\tabu@global@temp</code>	98, 155, 868, 878, 879, 918, 920
<code>\tabu@cell@j</code>	928	<code>\tabu@global@X</code>	155, 257, 264, 275, 284, 396, 400, 412, 454
<code>\tabu@cell@L</code>	963, 970, 1008	<code>\tabu@gobblespace</code>	64, 207, 568, 775
<code>\tabu@cell@l</code>	928	<code>\tabu@halignto</code>	117, 218, 231, 413, 486
<code>\tabu@cell@R</code>	979, 986, 1009	<code>\tabu@header</code>	448, 490, 502, 511
<code>\tabu@cell@r</code>	928	<code>\tabu@hfil</code>	1015, 1035, 1041
<code>\tabu@cellfont</code>	843, 846, 855, 923	<code>\tabu@hfill</code>	1016, 1042
<code>\tabu@celllalign</code>	856, 905, 923, 929, 937, 945, 953, 964, 972, 980	<code>\tabu@hfuzz</code>	186, 452, 504
<code>\tabu@celllalign</code>	988	<code>\tabu@hskip</code>	1017, 1043
<code>\tabu@cellleft</code>	595, 841, 846, 847, 853, 906, 923, 931, 933, 934, 939, 941, 942, 947, 949, 950, 953, 955, 956, 966, 968, 969, 974, 976, 977, 982, 984, 985, 988, 990, 991	<code>\tabu@ignorehfil</code>	932, 940, 948, 967, 975, 983, 1031
<code>\tabu@cellralign</code>	857, 902, 923, 932, 940, 948, 954, 967, 975, 983, 989	<code>\tabu@justify</code>	953, 955, 958, 961
<code>\tabu@cellright</code>	842, 854, 902, 923	<code>\tabu@lasttry</code>	393, 447, 450
<code>\tabu@cellrightfalse</code>	870, 881, 893	<code>\tabu@leaderfill</code>	716, 764, 803, 804
<code>\tabu@cellrighttrue</code>	886	<code>\tabu@linecolor</code>	767, 768, 774, 775, 779
<code>\tabu@cellskip</code>	996, 1014, 1020, 1021, 1022	<code>\tabu@linedash</code>	655, 722, 747
<code>\tabu@centering</code>	176, 1056	<code>\tabu@lineoff</code>	731, 734, 739, 742, 766
<code>\tabu@cline</code>	658, 659, 660	<code>\tabu@lineon</code>	722, 725, 741
<code>\tabu@cnt</code>	51, 354, 357, 392, 446, 512, 515	<code>\tabu@lines</code>	36, 87, 94, 636
<code>\tabu@collect</code>	415, 417	<code>\tabu@linestyle</code>	786, 787, 793
<code>\tabu@collectbody</code>	417	<code>\tabu@maybecolor</code>	630, 723, 729, 730, 733, 736, 743
<code>\tabu@color</code>	622, 633, 723, 726, 737, 751, 783	<code>\tabu@measuringfalse</code>	185, 437
<code>\tabu@colortblfalse</code>	49	<code>\tabu@measuringtrue</code>	251, 260, 280, 463, 488
<code>\tabu@colortbltrue</code>	49	<code>\tabu@message</code>	18, 19, 448, 451, 453, 471, 477, 478, 481, 482
<code>\tabu@colortest</code>	723, 767	<code>\tabu@message@arith</code>	451, 490
<code>\tabu@dashgap</code>	656, 731, 734, 745	<code>\tabu@message@reached</code>	453, 502
<code>\tabu@decimal</code>	556, 609	<code>\tabu@message@spread</code>	471, 497
<code>\tabu@defaultleaders</code>	707, 720, 795, 796	<code>\tabu@msg@align</code>	477, 481, 491, 492, 493, 494, 495, 498, 499, 500, 501, 503, 506
<code>\tabu@defleaders</code>	708, 709, 710, 715	<code>\tabu@naturalX</code>	54, 384, 469, 473, 501, 532, 538, 541, 543, 544
<code>\tabu@end@fbox</code>	1054, 1055	<code>\tabu@naturalX@max</code>	385, 469, 543, 544
<code>\tabu@endarray</code>	47, 154, 1071, 1077, 1086	<code>\tabu@naturalX@min</code>	386, 470, 546, 548
<code>\tabu@endenvir</code>	421, 441	<code>\tabu@nbcols</code>	51, 139, 687, 695, 700, 702, 703, 867, 868, 873, 874, 875, 877, 879, 888, 889, 890, 909
<code>\tabu@endTRIAL</code>	429	<code>\tabu@NC@list</code>	92, 105, 109
<code>\tabu@everycr</code>	253, 647, 648, 650, 651, 652, 653	<code>\tabu@NC@rewrite@X</code>	65, 97
<code>\tabu@everycr@tok</code>	127, 840, 858, 859	<code>\tabu@nestedfalse</code>	159
<code>\tabu@fbox</code>	182, 1047, 1053	<code>\tabu@nestedtrue</code>	157
<code>\tabu@firstcline</code>	35, 99, 645	<code>\tabu@next</code>	774, 775, 776, 778
<code>\tabu@firstclinefalse</code>	652, 690	<code>\tabu@nocline</code>	34, 41
<code>\tabu@firstclinetrue</code>	413, 646	<code>\tabu@nohfil</code>	1032, 1034, 1036, 1040
<code>\tabu@fix@arrayright</code>	48, 1066, 1096	<code>\tabu@norowfont</code>	33, 39
		<code>\tabu@nowrite</code>	129, 404

<code>\tabu@phantomline</code> .....	<a href="#">135</a> , <a href="#">253</a>	<code>\tabu@temp</code> .....	<a href="#">72</a> , <a href="#">110</a> , <a href="#">115</a> , <a href="#">118</a> , <a href="#">120</a> , <a href="#">121</a> , <a href="#">122</a> , <a href="#">125</a> , <a href="#">231</a> , <a href="#">250</a> , <a href="#">298</a> , <a href="#">369</a> , <a href="#">372</a> , <a href="#">373</a> , <a href="#">560</a> , <a href="#">563</a> , <a href="#">564</a> , <a href="#">579</a> , <a href="#">580</a> , <a href="#">581</a> , <a href="#">582</a> , <a href="#">583</a> , <a href="#">584</a> , <a href="#">585</a> , <a href="#">586</a> , <a href="#">587</a> , <a href="#">588</a> , <a href="#">589</a> , <a href="#">590</a> , <a href="#">591</a> , <a href="#">592</a> , <a href="#">593</a> , <a href="#">594</a> , <a href="#">595</a> , <a href="#">596</a> , <a href="#">597</a> , <a href="#">606</a> , <a href="#">607</a> , <a href="#">608</a> , <a href="#">613</a> , <a href="#">617</a> , <a href="#">618</a> , <a href="#">623</a> , <a href="#">633</a> , <a href="#">767</a> , <a href="#">769</a> , <a href="#">770</a> , <a href="#">771</a> , <a href="#">772</a> , <a href="#">774</a> , <a href="#">775</a>
<code>\tabu@prep@TRIAL</code> .....	<a href="#">268</a> , <a href="#">291</a> , <a href="#">376</a>	<code>\tabu@to</code> .....	<a href="#">208</a> , <a href="#">213</a>
<code>\tabu@prepnext@tok</code> .....	<a href="#">31</a> , <a href="#">183</a> , <a href="#">862</a>	<code>\tabu@TRIAL</code> .....	<a href="#">422</a> , <a href="#">429</a>
<code>\tabu@preptokenleft</code> .....	<a href="#">884</a> , <a href="#">905</a> , <a href="#">907</a>	<code>\tabu@TRIAL@hook</code> .....	<a href="#">407</a> , <a href="#">409</a> , <a href="#">410</a>
<code>\tabu@preptokenright</code> ....	<a href="#">882</a> , <a href="#">894</a> , <a href="#">900</a> , <a href="#">904</a>	<code>\tabu@trivlist</code> .....	<a href="#">179</a> , <a href="#">1063</a>
<code>\tabu@printdecimal</code> .....	<a href="#">602</a> , <a href="#">605</a> , <a href="#">611</a>	<code>\tabu@usetabu</code> .....	<a href="#">72</a> , <a href="#">81</a>
<code>\tabu@pushbegins</code> .....	<a href="#">419</a> , <a href="#">428</a>	<code>\tabu@usetabuX</code> .....	<a href="#">251</a> , <a href="#">252</a> , <a href="#">254</a>
<code>\tabu@raggedleft</code> .....	<a href="#">178</a> , <a href="#">1058</a>	<code>\tabu@verb</code> .....	<a href="#">806</a> , <a href="#">810</a>
<code>\tabu@raggedright</code> .....	<a href="#">177</a> , <a href="#">1060</a>	<code>\tabu@vline</code> .....	<a href="#">614</a> , <a href="#">617</a> , <a href="#">634</a> , <a href="#">642</a>
<code>\tabu@removehfil</code> <a href="#">929</a> , <a href="#">937</a> , <a href="#">945</a> , <a href="#">964</a> , <a href="#">972</a> , <a href="#">980</a> , <a href="#">1014</a>		<code>\tabu@write</code> .....	<a href="#">134</a> , <a href="#">460</a>
<code>\tabu@restoreeverycr</code> .....	<a href="#">648</a> , <a href="#">649</a>	<code>\tabu@X@cols</code> .....	<a href="#">51</a> , <a href="#">273</a> , <a href="#">274</a> , <a href="#">276</a> , <a href="#">278</a> , <a href="#">281</a> , <a href="#">283</a> , <a href="#">285</a> , <a href="#">289</a> , <a href="#">382</a> , <a href="#">383</a> , <a href="#">384</a> , <a href="#">536</a> , <a href="#">537</a> , <a href="#">540</a> , <a href="#">547</a> , <a href="#">549</a>
<code>\tabu@rewrite@X</code> .....	<a href="#">65</a> , <a href="#">71</a> , <a href="#">244</a> , <a href="#">397</a>	<code>\tabu@X@sum</code> .....	<a href="#">57</a> , <a href="#">258</a> , <a href="#">265</a> , <a href="#">277</a> , <a href="#">278</a> , <a href="#">286</a> , <a href="#">289</a> , <a href="#">296</a> , <a href="#">300</a> , <a href="#">357</a> , <a href="#">362</a> , <a href="#">369</a> , <a href="#">457</a> , <a href="#">458</a> , <a href="#">494</a> , <a href="#">496</a> , <a href="#">546</a> , <a href="#">548</a>
<code>\tabu@rewrite@Xnested</code> .....	<a href="#">249</a> , <a href="#">397</a>	<code>\tabu@Xalign</code> <a href="#">330</a> , <a href="#">331</a> , <a href="#">332</a> , <a href="#">335</a> , <a href="#">336</a> , <a href="#">337</a> , <a href="#">343</a> , <a href="#">347</a>	
<code>\tabu@rewrite@Xrestore</code> .....	<a href="#">71</a> , <a href="#">249</a>	<code>\tabu@Xarg</code> .....	<a href="#">244</a> , <a href="#">250</a> , <a href="#">294</a>
<code>\tabu@rewrite@Xspread</code> .....	<a href="#">245</a> , <a href="#">271</a>	<code>\tabu@Xc@ef</code> .....	<a href="#">354</a> , <a href="#">356</a> , <a href="#">359</a>
<code>\tabu@rewrite@Xto</code> .....	<a href="#">246</a> , <a href="#">255</a>	<code>\tabu@Xcoef</code> .....	<a href="#">333</a> , <a href="#">339</a> , <a href="#">353</a> , <a href="#">355</a>
<code>\tabu@rewritefirst</code> .....	<a href="#">86</a> , <a href="#">101</a> , <a href="#">102</a> , <a href="#">103</a> , <a href="#">104</a> , <a href="#">106</a> , <a href="#">107</a> , <a href="#">187</a> , <a href="#">234</a>	<code>\tabu@Xcolspec</code> .....	<a href="#">327</a> , <a href="#">328</a> , <a href="#">329</a> , <a href="#">348</a> , <a href="#">352</a>
<code>\tabu@rewritefirst_(new_column_type)</code> ..	<a href="#">86</a>	<code>\tabu@Xfinish</code> .....	<a href="#">412</a> , <a href="#">442</a>
<code>\tabu@rewritelast</code> .....	<a href="#">89</a> , <a href="#">93</a> , <a href="#">100</a> , <a href="#">108</a> , <a href="#">110</a> , <a href="#">111</a> , <a href="#">112</a> , <a href="#">113</a> , <a href="#">114</a> , <a href="#">126</a>	<code>\tabu@xfootnotetext</code> .....	<a href="#">171</a> , <a href="#">241</a>
<code>\tabu@rewritelast_(new_column_type)</code> ..	<a href="#">108</a>	<code>\tabu@xleaders</code> .....	<a href="#">663</a> , <a href="#">677</a> , <a href="#">705</a> , <a href="#">707</a> , <a href="#">716</a> , <a href="#">720</a> , <a href="#">751</a> , <a href="#">790</a> , <a href="#">791</a> , <a href="#">795</a>
<code>\tabu@row@font</code> .....	<a href="#">838</a> , <a href="#">839</a>	<code>\tabu@Xmath</code> .....	<a href="#">303</a> , <a href="#">338</a> , <a href="#">364</a>
<code>\tabu@rowfont</code> .....	<a href="#">33</a> , <a href="#">184</a> , <a href="#">838</a>	<code>\tabu@Xparse</code> <a href="#">304</a> , <a href="#">307</a> , <a href="#">308</a> , <a href="#">338</a> , <a href="#">340</a> , <a href="#">346</a> , <a href="#">351</a> , <a href="#">358</a>	
<code>\tabu@rowfont@reset</code> .....	<a href="#">848</a> , <a href="#">851</a>	<code>\tabu@Xparsespace</code> .....	<a href="#">308</a> , <a href="#">334</a>
<code>\tabu@sanitizetext</code> .....	<a href="#">156</a> , <a href="#">181</a> , <a href="#">805</a>	<code>\tabu@Xtest</code> .....	<a href="#">307</a> , <a href="#">309</a> , <a href="#">342</a>
<code>\tabu@save@decl</code> .....	<a href="#">32</a> , <a href="#">96</a> , <a href="#">189</a> , <a href="#">197</a>	<code>\tabucline</code> .....	<a href="#">6</a> , <a href="#">6</a> , <a href="#">34</a> , <a href="#">42</a> , <a href="#">655</a>
<code>\tabu@saved</code> .....	<a href="#">223</a> , <a href="#">230</a> , <a href="#">816</a> , <a href="#">825</a>	<code>\tabucline@warn</code> .....	<a href="#">717</a> , <a href="#">721</a> , <a href="#">737</a>
<code>\tabu@savedpreamble</code> .....	<a href="#">377</a> , <a href="#">414</a>	<code>\tabucolX</code> .....	<a href="#">57</a> , <a href="#">249</a> , <a href="#">297</a> , <a href="#">298</a> , <a href="#">369</a> , <a href="#">384</a> , <a href="#">390</a> , <a href="#">455</a> , <a href="#">457</a> , <a href="#">459</a> , <a href="#">462</a> , <a href="#">470</a> , <a href="#">473</a> , <a href="#">487</a> , <a href="#">491</a> , <a href="#">500</a> , <a href="#">503</a> , <a href="#">522</a> , <a href="#">827</a>
<code>\tabu@saveerr</code> .....	<a href="#">68</a> , <a href="#">70</a> , <a href="#">832</a>	<code>\tabucolX@align</code> .....	<a href="#">301</a> , <a href="#">344</a> , <a href="#">346</a> , <a href="#">370</a>
<code>\tabu@savetok</code> .....	<a href="#">909</a> , <a href="#">911</a>	<code>\tabucolX@do</code> .....	<a href="#">326</a> , <a href="#">360</a> , <a href="#">375</a>
<code>\tabu@savetok</code> <a href="#">882</a> , <a href="#">884</a> , <a href="#">886</a> , <a href="#">890</a> , <a href="#">894</a> , <a href="#">908</a> , <a href="#">910</a> , <a href="#">922</a>		<code>\tabucolX@spec</code> .....	<a href="#">66</a> , <a href="#">249</a> , <a href="#">250</a> , <a href="#">297</a> , <a href="#">302</a> , <a href="#">349</a> , <a href="#">351</a> , <a href="#">361</a> , <a href="#">365</a> , <a href="#">367</a> , <a href="#">372</a> , <a href="#">373</a>
<code>\tabu@savewarn</code> .....	<a href="#">814</a> , <a href="#">815</a> , <a href="#">823</a> , <a href="#">824</a> , <a href="#">832</a>	<code>\tabudecimal</code> .....	<a href="#">8</a> , <a href="#">180</a> , <a href="#">555</a>
<code>\tabu@saveX</code> .....	<a href="#">378</a> , <a href="#">406</a> , <a href="#">813</a>	<code>\tabudefaultright</code> <a href="#">20</a> , <a href="#">24</a> , <a href="#">158</a> , <a href="#">161</a> , <a href="#">262</a> , <a href="#">282</a> , <a href="#">398</a>	
<code>\tabu@scandecimal</code> <a href="#">558</a> , <a href="#">560</a> , <a href="#">561</a> , <a href="#">568</a> , <a href="#">573</a> , <a href="#">576</a>		<code>\tabuDisableCommands</code> .....	<a href="#">409</a>
<code>\tabu@settarget</code> .....	<a href="#">148</a> , <a href="#">198</a>	<code>\tabular</code> .....	<a href="#">233</a>
<code>\tabu@setup</code> .....	<a href="#">148</a> , <a href="#">150</a>	<code>\tabulinesstyle</code> .....	<a href="#">7</a> , <a href="#">786</a>
<code>\tabu@setup@TRIAL</code> .....	<a href="#">380</a> , <a href="#">396</a> , <a href="#">408</a>	<code>\tabuvline@arg</code> .....	<a href="#">612</a> , <a href="#">615</a>
<code>\tabu@skipdecimal</code> .....	<a href="#">561</a> , <a href="#">567</a> , <a href="#">569</a> , <a href="#">600</a>	<code>\tabuvline@argi</code> .....	<a href="#">618</a> , <a href="#">621</a> , <a href="#">628</a>
<code>\tabu@spread</code> .....	<a href="#">209</a> , <a href="#">214</a>	<code>\tabuvline@argii</code> .....	<a href="#">626</a> , <a href="#">629</a> , <a href="#">631</a>
<code>\tabu@spread@endhbox</code> .....	<a href="#">521</a>	<code>\tabuvline@argiii</code> .....	<a href="#">618</a> , <a href="#">632</a> , <a href="#">635</a>
<code>\tabu@spread@startpbox</code> .....	<a href="#">388</a> , <a href="#">521</a>	<code>\tabuvline@rewrite</code> .....	<a href="#">612</a> , <a href="#">641</a>
<code>\tabu@spreadarith</code> .....	<a href="#">434</a> , <a href="#">445</a>	<code>\tabuvlines@find</code> .....	<a href="#">640</a> , <a href="#">644</a>
<code>\tabu@spreadfalse</code> .....	<a href="#">185</a> , <a href="#">432</a>	<code>\TMP@EnsureCode</code> .....	<a href="#">7</a> , <a href="#">13</a> , <a href="#">14</a> , <a href="#">15</a> , <a href="#">16</a> , <a href="#">17</a>
<code>\tabu@spreadheader</code> .....	<a href="#">497</a> , <a href="#">516</a>	<code>\toks</code> ...	<a href="#">191</a> , <a href="#">192</a> , <a href="#">194</a> , <a href="#">195</a> , <a href="#">902</a> , <a href="#">905</a> , <a href="#">906</a> , <a href="#">912</a>
<code>\tabu@spreadtarget</code> .....	<a href="#">54</a> , <a href="#">281</a> , <a href="#">288</a> , <a href="#">472</a> , <a href="#">473</a> , <a href="#">474</a> , <a href="#">475</a> , <a href="#">478</a> , <a href="#">498</a>	<code>\tracingtabu</code> .....	<a href="#">12</a>
<code>\tabu@spreadtrue</code> .....	<a href="#">214</a>	<code>\tracingtabu_(debugshow_option)</code> .....	<a href="#">18</a>
<code>\tabu@start</code> .	<a href="#">662</a> , <a href="#">667</a> , <a href="#">671</a> , <a href="#">697</a> , <a href="#">698</a> , <a href="#">700</a> , <a href="#">701</a>		
<code>\tabu@start@stop</code> .....	<a href="#">695</a> , <a href="#">696</a> , <a href="#">704</a>		
<code>\tabu@startpboxORI</code> .....	<a href="#">387</a> , <a href="#">433</a> , <a href="#">527</a>		
<code>\tabu@startstop</code> .....	<a href="#">661</a> , <a href="#">695</a>		
<code>\tabu@stop</code> <a href="#">662</a> , <a href="#">674</a> , <a href="#">679</a> , <a href="#">682</a> , <a href="#">697</a> , <a href="#">699</a> , <a href="#">702</a> , <a href="#">703</a>			
<code>\tabu@tabudecimal</code> .....	<a href="#">180</a> , <a href="#">555</a>		
<code>\tabu@target</code> ..	<a href="#">54</a> , <a href="#">118</a> , <a href="#">199</a> , <a href="#">215</a> , <a href="#">217</a> , <a href="#">231</a> , <a href="#">261</a> , <a href="#">262</a> , <a href="#">266</a> , <a href="#">281</a> , <a href="#">282</a> , <a href="#">287</a> , <a href="#">390</a> , <a href="#">452</a> , <a href="#">454</a> , <a href="#">457</a> , <a href="#">472</a> , <a href="#">474</a> , <a href="#">475</a> , <a href="#">477</a> , <a href="#">478</a> , <a href="#">481</a> , <a href="#">482</a> , <a href="#">486</a> , <a href="#">487</a> , <a href="#">493</a> , <a href="#">495</a> , <a href="#">525</a> , <a href="#">532</a> , <a href="#">818</a> , <a href="#">828</a>		

<code>\ttfamily</code> .....	809	<code>\vfill</code> .....	524
<b>U</b>		<code>\vskip</code> .....	524, 666
<code>\unhbox</code> .....	535	<code>\vss</code> .....	524
<code>\unkern</code> .....	1020	<code>\vtop</code> .....	553
<code>\unskip</code> .....	1025, 1114	<b>W</b>	
<code>\usetabu</code> .....	10, 67, 84, 85, 88, 91, 298, 836	<code>\write</code> .....	129, 134, 404, 460
<code>\usetabu_(new_column_type)</code> .....	67	<b>X</b>	
<b>V</b>		<code>\xleaders</code> .....	716, 751
<code>\value</code> .....	399	<b>Z</b>	
<code>\vbox</code> .....	429	<code>\zap@space</code> .....	781
<code>\verbatim</code> .....	8, 156, 181		
<code>\vfil</code> .....	524		