

tabu

tabu and longtabu

Flexible L^AT_EX tabulars

FC

2010/12/18 – version 1.7

Abstract

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

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

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

`tabu` requires ϵ -T_EX and the standard package `array.sty`. Natural widths of columns are computed by (but not printed by) the code of `varwidth` by D. Arseneau (for the cases of X columns in `tabu spread` and *negative width coefficients*¹). Finally `longtabu` is based on `longtable`.

Contents

Summary of the features provided by <code>tabu</code>	3
1 The <code>tabu</code> environment	4
1.1 <code>tabu</code> , <code>tabu to</code> and <code>tabu spread</code>	4
1.2 <code>longtabu</code> , <code>longtabu to</code> and <code>longtabu spread</code>	4
1.3 <code>tabu</code> X columns	5
1.3.1 X columns with “ <code>tabu spread</code> ”	
1.3.2 Negative width coefficients for X columns	
1.3.3 Multicolumn in <code>tabu</code>	
1.3.4 Nesting <code>tabus</code> with X columns	
1.4 Lines inside <code>tabu</code>	7
1.4.1 First important remarks	
1.4.2 Vertical lines: <code> </code> has an optional parameter	
1.4.3 More style for horizontal lines: <code>\tabucline</code>	
1.5 <code>tabu</code> in math mode	9
1.6 Verbatim inside <code>tabu</code> with X columns.....	10
1.7 Printing numbers inside <code>tabu</code> with <code>numprint</code> and <code>siunitx</code>	10
1.7.1 Just make it easy !	
1.7.2 You should know how it works... ..	
1.8 <code>delarray</code> shortcuts.....	11
2 Saving and restoring a <code>tabu</code> : <code>\savetabu</code> , <code>\usetabu</code> and <code>\preamble</code>	12
3 Modifying the font and the alignment in one row: <code>\rowfont</code>	13
4 Differences between <code>tabu</code> , <code>tabular</code> , <code>tabularx</code> and <code>longtable</code>	13
4.1 Paragraph indentation	13
4.2 Custom environments.....	14
4.3 Inversion of tokens	14
4.4 Improved process for rewriting columns (<i>for keen readers</i>)	14
5 The package options	15

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

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

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

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

5.1	The <code>debugshow</code> package option	15
5.2	The <code>linegoal</code> package option	16
5.3	The <code>delarray</code> package option	16
5.4	The <code>light</code> package option	16
6	Corrections of some bugs (<i>available only inside tabu</i>)	16
6.1	Correction for <code>colortbl</code> and <code>arydshln</code> : compatibility with <code>delarray</code>	16
6.2	Correction for <code>arydshln</code> : @ columns	17
7	To do for even better tabus	17
8	IMPLEMENTATION	18
8.1	Identification, requirements and options	18
8.2	Some constants	19
8.3	<code>\tabu</code> , <code>\endtabu</code> , <code>\longtabu</code> and <code>\endlontabu</code>	22
8.4	Footnotes inside <code>tabu</code>	25
8.5	Flow chart of expansion	27
8.5.1	<i>General case: <code>tabu</code>, <code>tabu to</code> and <code>tabu spread</code></i>	
8.5.2	<i><code>tabu to</code> with <code>X</code> column</i>	
8.5.3	<i><code>tabu spread</code> with <code>X</code> column</i>	
8.6	<code>tabu X</code> column definition	29
8.7	Before trials to reach the target: setup after <code>\@mkpream</code>	31
8.8	One trial after another	32
8.9	Algorithms: Measuring the <code>tabu</code>	34
8.9.1	<i>The arithmetic of <code>X</code> columns</i>	
8.9.2	<i>The arithmetic of <code>X</code> columns for <code>tabu spread</code></i>	
8.9.3	<i>Reporting in the <code>.log</code> file (<code>debugshow</code> option)</i>	
8.9.4	<i>Measuring the natural width for <code>tabu spread</code> and negativ width-coefficients</i>	
8.10	<code>\everycr</code> setup and <code>\tabuphantomline</code>	38
8.11	Lines inside <code>tabu</code>	39
8.11.1	<i>Vertical lines</i>	
8.11.2	<i>Horizontal lines: <code>\tabucline</code></i>	
8.12	Verbatim inside <code>tabu</code> with <code>X</code> columns	43
8.13	Numbers in <code>tabu</code>	43
8.13.1	<i><code>\tabudecimal</code></i>	
8.14	<code>\savetabu</code>	44
8.15	<code>\rowfont</code>	45
8.15.1	<i>Setting font and alignment specification</i>	
8.15.2	<i>Preparing stuff to be able to use <code>\rowfont</code></i>	
8.15.3	<i>Neutralisation of glues and alignment modification</i>	
8.16	Utilities	50
8.16.1	<i><code>tabu \fbox</code></i>	
8.16.2	<i><code>\centering</code>, <code>\raggedright</code>, <code>\raggedleft</code></i>	
8.17	Corrections	51
8.17.1	<i><code>delarray</code> comptability fix for <code>colortbl</code> and <code>arydshln</code></i>	
8.17.2	<i><code>arydshln</code> @ columns</i>	
8.17.3	<i><code>arydshln</code>, colors without <code>colortbl</code> and empty <code>p</code> columns</i>	
9	References	53
10	History	53
	[2010/12/18 v1.7]	53
	[2010/12/07 v1.5]	53
	[2010/11/22 v1.4]	53
	[2010/11/18 v1.3]	54
	[2010/11/15 v1.2]	54
	[2010/10/28 v1.1]	54
11	Index	54

Summary of the features provided by **tabu**

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

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

1 The tabu environment

1.1 tabu, tabu to and tabu spread

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

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

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

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

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

1.2 longtabu, longtabu to and longtabu spread

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

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

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

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

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

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

1.3 tabu X columns

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

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

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

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

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

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

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

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

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

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

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

X columns with “`tabu spread`”

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

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

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

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

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

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

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

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

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

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

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

"Like the air we breathe, Sherlock Holmes is everywhere. His pipe-smoking, deer stalked image peers at us from ads in Yellow Pages, to signs for neighbourhood crime-watch; from billboards to the classroom; from film and television to the public library, and now over the Internet. He long ago transcended the boundaries of 19th Century London² to become an international best-seller and has been accepted as part of British folklore. Holmes is alive to millions."

There the text was too long, and `tabu spread` behaves as if you didn’t give it a target.

The result of this example is the same as if one had written `\begin{tabu}to\linewidth.`

Sherlock Holmes

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

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

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

Negativ width coefficients for X columns

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

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

Multicolumn in tabu

`\tabuphantomline`

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

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

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

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

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

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

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

Nesting tabus with X columns

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

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

1.4 Lines inside tabu

First important remarks

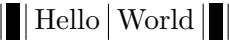

The features provided in this section are quite experimental: they are not generally taken for good typography. You can use `tabu` with package `booktabs` for example, which provides properly designed commands for horizontal rules in tabulars. `arydshln` is pretty good too, but it modifies a huge amount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

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

Vertical lines: | has an optional parameter

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

Example:

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

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

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

More style for horizontal lines: \tabucline

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

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

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

Any parameter can be omitted.

<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 gap 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:

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

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





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

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

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

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

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

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

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

1.5 tabu in math mode

On the left, you can see the famous Maxwell-Lorentz equations for electromagnetic field in vacuum, published in 1873.	$\left\{\begin{array}{l} \operatorname{div} \vec{E} = \frac{\rho}{\epsilon_0} \\ \operatorname{div} \vec{B} = 0 \\ \operatorname{rot} \vec{E} = -\frac{\partial \vec{B}}{\partial t} \\ \operatorname{rot} \vec{B} = \mu_0 \vec{j} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \end{array}\right.$
---	---

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
$\begin{array}{ c c } \hline \alpha & \beta \\ \hline \gamma & \delta \\ \hline \end{array}$	$\begin{array}{ c c } \hline \alpha & \beta \\ \hline \gamma & \delta \\ \hline \end{array}$	$\begin{array}{ c c } \hline \alpha & \beta \\ \hline \gamma & \delta \\ \hline \end{array}$

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

Vertical lines inside the tabular preamble gives:

$\begin{array}{ c c } \hline \alpha & \beta \\ \hline \gamma & \delta \\ \hline \end{array}$
--

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

1.6 Verbatim inside tabu with X columns

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

`\verbatim{<general text>}`

To get round this limitation, `tabu` provides the command `\verbatim` which allows to put some pieces of verbatim text inside a `tabu`: it is based on ϵ -T_EX `\scantokens` primitive and is defined only inside the `tabu` environment. `\verbatim` has some other limitations you must know:

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

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

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

1.7 Printing numbers inside tabu with numprint and siunitx

Just make it easy !

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

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

`\tabudecimal{⟨user-macro⟩}`

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

1.8 delarray shortcuts

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

with overfull hboxes
(15.8pt and 16.1pt two wide)
for tabularx

(This is `tabu` with `delarray` shortcut for parenthesis around)

(This is `tabularx` with `delarray` shortcut for parenthesis around)

{ This is `tabu` with `delarray` shortcut for curly brackets around }

{ This is `tabularx` with `delarray` shortcut for curly brackets around }

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

`\savetabu{⟨user-name⟩}`

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

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

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

`\usetabu{⟨user-name⟩}`

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

`\usetabu` does not work with [longtabu](#).

`\usetabu` locally restores:

- the preamble³.
- the vertical position `[c]`, `[b]` or `[t]`, unless another position is specified.
- the target width of the `tabu` in points: the saved target width does not contain any control sequence: it is fixed and stored in points.
- the width of `tabu` X columns: those widths are not calculated any more, even in the case of negativ coefficients and X columns are directly transformed into `p`, `m` or `b` columns of the same widths as the ones that where calculated at the time of `\savetabu`
- `\tabcolsep` (or `\arraycolsep` in math mode) and `\arraystretch`
- `\arrayrulewidth`, `\doublerulesep`, `\extrarowheight` and `\extratabsurround`
- `\minrowclearance`, `\arrayrulecolor` and `\doublerulesepcolor` (package `colortbl`)

Example:

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

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

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

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

If one day you use `tabu`, you will have the idea to restore a `tabu` while modifying its target, or adding new columns... `\savetabu` and `\usetabu` have not been thought for this purpose, and you may have unexpected results. `\usetabu` is a help to get several tabulars of exactly the same shape, same target, same preamble. The only parameter that can be changed is the optional vertical position parameter for the whole tabular.

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

```
\preamble{⟨user-name⟩}
```

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

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

Any other tabular parameters are not restored.

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

`\preamble` works exactly as if you defined a [custom environment](#) for `tabu`.

`\preamble` works with `longtabu`.

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

```
\rowfont[⟨alignment⟩]{font specification}
```

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

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

<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.

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

This	Is
tabu	package for tabu and longtabu

4 Differences between `tabu`, `tabular` `tabularx` and `longtable`

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^AT_EX 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`.

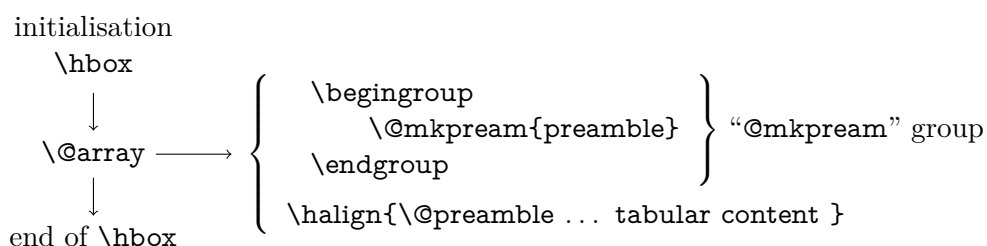
This rather strange inversion of tokens may be justified in math mode (otherwise, errors may occur) but not in text mode in our opinion. Inside a `tabu` environment, when not in math mode, the tokens are not reversed and you get the intuitively expected result:

`| 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 (*for keen readers*)

Any tabulars that do not split accross pages are made with the following process:



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

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

First the rewriting process:

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

Second the building of the `\halign` preamble:

The “rewritten preamble” is parsed and transformed in a preamble for the T_EX primitive `\halign`. The result is stored into the `\@preamble` macro.

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

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

Inside `tabular`:

- 1) Rewrite `S`: not found because inside `{...}`
- 2) Rewrite `*`
- 3) Rewrite `n` column defined by package `numprint`
Then the ‘n’ in `green` is rewritten → problem

```

\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}

```

Inside `tabu`:

- 1) Rewrite `*`
- 2) Rewrite `|` (there is none here)

go back

- 3) Rewrite `*`
- 4) Rewrite `|`
- 5) Rewrite `S`
- 6) Rewrite `n` → not found because `S` was rewritten before, according to `siunitx` definition.

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

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

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

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

it is not recommended no nest such constructions like:

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

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

Well, just avoid to nest new column type definitions !

5 The package options

5.1 The `debugshow` package option

```

\tracingtabu
\tracingtabu = 2

```

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

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

Typical information in the `.log` file:

```

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

```

What does it mean?

- 1) The first attempt was performed with `X=386.67296pt`
The `tabu` width (797.34592pt) exceeded the target by 410.67296pt.

Thus X has been updated: $410.67296\text{pt} / 2 = 205.33649\text{pt}$ and then:
 $X = 386.67296\text{pt} - 205.33649\text{pt} = 181.33647\text{pt}$

- 2) The second attempt lead to a `tabu` width of 386.67294pt : the target is reached.
 The final width of each X column is the product of `tabu` X by its width coefficient.

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

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_TE_X, otherwise, a warning is displayed and the `linegoal` option has no effect: the default target remains `\linewidth`. `\linegoal` works with pdf_TE_X 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`

In any case, specifying the target overwrites the default: `\begin{tabu}` to `\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`. As a result, `delarray` shortcuts for delimiters around a tabular does not work if `colortbl` and/or `arydshln` are loaded.

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

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

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 as far as rules are concerned.

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

7 To do for even better `tabu`

- ➡ Make double `\tabucline` with compatibility with `colortbl` `\doublerulesepcolor`
- ➡ Multiple `\tabucline` between different columns: extended specs:
`\tabucline[line spec]{start-stop,start-stop}[line spec]{start-stop} ...`
- ➡ Presently, `longtabu` with `X` columns works only if `\LTchunksize` is greater than the number of rows. I compiled a `longtabu` of 56 pages on my PC with `\LTchunksize = 2000` without problem. Presently `\LTchunksize` is set to 10 000 during trials when `longtabu` contains `X` columns.
- ➡ Make `\tabucline` work with page breaks (one line on the top of the page, one line on the bottom of the previous).
- ➡ An environment “`calstabu`” based on the `cals` package but with the syntax of `tabular` if possible...

8 Implementation

8.1 Identification, requirements and options

The package namespace is `tabu@`.

```

1 \*package
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{tabu}[2010/12/18 v1.7 - flexible LaTeX tabulars (FC)]
4 \RequirePackage{array}[2008/09/09]
5 \RequirePackage{varwidth}[2009/03/30]
6 \let\DEBUG\@gobble%\message

```

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

```

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

```

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

```

20 \let\tabu@message \@gobble
21 \let\tabu@message@save \@gobble
22 \let\tabu@debug \@gobble
23 \def\tracingtabu{%
24   \let\tabu@message \message
25   \let\tabu@message@save \tabu@tracing@save
26   \@ifnextchar={\tabu@tracing}{}}
27 \def\tabu@tracing{\begingroup\afterassignment\tabu@tr@cing\count@}
28 \def\tabu@tr@cing{\expandafter\endgroup\ifnum\count@>\@ne
29   \let\tabu@debug\message\else\let\tabu@debug\@gobble\fi}
30 \DeclareOption{debugshow}{\AtEndOfPackage\tracingtabu}

```

`linegoal` (package option)

```

31 \let\tabubefaulttarget \linewidth
32 \DeclareOption{linegoal}{%
33   \AtEndOfPackage{%
34     \RequirePackage{linegoal}[2010/12/07]%
35     \let\tabubefaulttarget \linegoal% \linegoal is \linewidth if not pdfTeX
36 }}

```

`delarray` (package option)

```

37 \DeclareOption{delarray}{%
38   \AtEndOfPackage{\RequirePackage{delarray}}%
39 }

```

`light` (package option)

```

40 \DeclareOption{light}{%
41   \AtEndOfPackage{%
42     \let\tabu@prepnxt@tok \prepnxt@tok \let\tabu@save@decl \save@decl
43     \let\tabu@rowfont \tabu@norowfont \let\tabu@cline \tabu@nocline

```

```

44 \let\tabu@firstcline \relax \let\tabu@lines \relax
45 \let\tabu@setup@everycr\relax
46 }
47 }
48 \def\tabu@norowfont{\PackageError{tabu}
49 {\string\rowfont\space is not available with option ‘light’}\@ehd}
50 \def\tabu@nocline{\PackageError{tabu}
51 {\string\tabucline\space is not available with option ‘light’}\@ehd}

52 \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.

```

53 \AtBeginDocument{%
54 \ifundefined{NC@rewrite@X}{\newcolumntype{X}{}{}}{% new column X if not exists
55 \expandafter\in@ \expandafter\@arrayright\expandafter{\endarray}%
56 \ifin@ \let\tabu@endarray \endarray
57 \else \tabu@fix@arrayright \fi % <fix for colortbl & arydshln (delarray)>
58 \ifpackageloaded{colortbl} \tabu@colortbltrue \tabu@colortblfalse
59 \ifpackageloaded{arydshln}{%
60 {\let\tabu@adl@fix \relax \let\tabu@adl@endTRIAL \@empty}%
61 \ifpackageloaded{longtable}{\let\longtabu \tabu@nolongtabu}%
62 }

```

8.2 Some constants

`\tabu@cnt` Used in `\tabu@arith` (the number of trials) and `\tabu@prepnext@tok` (for `\rowfont`).

`\tabu@nbcols` A counter that save the number of columns of the `tabu`.

`\tabu@X@cols` Used only when `tabu X` columns are used with “`tabu spread`”.

```

63 \newcount \tabu@cnt
64 \newcount \tabu@nbcols
65 \newcount \tabu@X@cols
66 \newcount \tabu@start
67 \newcount \tabu@stop
68 \newcount \tabu@nested

```

`\tabu@target` Stores the `tabu` target (either “to” or “spread”).

`\tabu@spreadtarget` Used only when `tabu X` columns are used with “`tabu spread`”.

```

\tabu@naturalX
69 \newdimen \tabu@target
70 \newdimen \tabu@spreadtarget
71 \newdimen \tabu@naturalX

```

`\tabu@colX` The dimen corresponding to the preamble token `X[1]`: the standard width of X columns.

`\tabu@X@sum` Stores the sum of all width coefficients for X columns.

`\tabu@colX@error` In case of X columns too narrow.

```

72 \newdimen \tabu@colX
73 \newdimen \tabu@X@sum
74 \newdimen \tabu@colX@error \tabu@colX@error=1em

```

<code>\iftabu@measuring</code>	<p>This switch is set to <code>true</code> by <code>\tabu@arith</code> if the trial did not reached the target.</p> <p>It is also temporarily set to <code>true</code> when the first <code>X</code> 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 <code>X</code> column found actually triggers some special setup to be expanded before <code>\halign</code> (see the flow chart...)</p>
<code>\iftabu@spread</code>	A switch whether “ <code>tabu spread</code> ” is used or not.
<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>).
<code>\iftabu@negcoef</code>	<p>A switch set to <code>true</code> in case of negativ coef (natural width if less than <code>X[coef]</code>).</p> <pre> 75 \newif \iftabu@measuring 76 \newif \iftabu@spread 77 \newif \iftabu@firstcline 78 \newif \iftabu@negcoef </pre>
<code>\tabu@box</code>	<p>Stores the whole <code>tabu</code> when an attempt to adjust <code>X</code> columns is performed.</p> <p>It is also used by <code>\tabucline</code> to save the <code>\@arstrutbox</code> when inserting a horizontal line.</p> <pre> 79 \newsavebox \tabu@box 80 \chardef \tabu@arstrutbox=\@arstrutbox 81 \let\tabu@vcenter \vcenter </pre>
<code>\tabu@gobblespace</code>	A macro which is needed when scanning tokens with <code>\futurelet</code> .
<code>\tabu@gobbleopt</code>	<p>A macro that gobbles an optional argument.</p> <pre> 82 \def\tabu@gobblespace#1 {#1} 83 \def\tabu@gobbleopt#1{\@ifnextchar [{\tabu@gobblept{#1}}{#1}}% 84 \def\tabu@gobblept#1[#2]{#1} </pre>
<code>\tabu@save@decl</code>	<p>No inversion on tokens in the <code>tabu</code> preamble, when not in math mode.</p> <pre> 85 \def\tabu@save@decl{% 86 \toks\count@ = \expandafter{\the\toks\expandafter\count@\@nextchar}% 87 }% \tabu@save@decl </pre>
<code>\tabu@NC@rewrite@X</code>	<p>This is the rewrite macro for <code>tabu X</code> columns. Such a column has an optional argument: the width coefficient for the <code>tabu X</code> column whose default value is 1, and may be some alignments parameters. The coefficient is used in the expression: <code>p{\dimexpr<coef>\tabucolX}</code></p> <pre> 88 \newcommand\tabu@NC@rewrite@X[1] [] {\tabu@rewrite@X{#1}% 89 \let\@halignto \relax 90 \expandafter \NC@find \tabucolX\rw} </pre> <p>The next part of the definition (<code>\tabu@rewrite@X</code>) can be found page 29.</p>
<code>\tabu@newcolumnntype</code>	<p>A helper macro to create local new column types for <code>tabu</code>.</p> <pre> 91 \def\tabu@newcolumnntype #1{% 92 \expandafter\def\csname NC@find@\string#1\endcsname##1#1{\NC@{##1}}% 93 \expandafter\def\csname NC@rewrite@\string#1\endcsname% 94 }% \tabu@newcolumnntype </pre>
<code>\usetabu</code> (new column type)	<p><code>\usetabu</code> is defined as a <code>tabu</code> new column type: loaded only inside the <code>\@mkpream</code> group inside the <code>tabu</code> environment.</p> <pre> 95 \tabu@newcolumnntype \usetabu #1{% 96 \ifx\\#1\\\tabu@saveerr{}\else 97 \ifundefined\tabu@saved@\string#1 98 {\tabu@saveerr{#1}} 99 {\let\tabu@rewrite@X \tabu@rewrite@Xrestore 100 \csname tabu@saved@\string#1\expandafter\endcsname\expandafter\z@}% 101 \fi 102 }% \NC@rewrite@\usetabu </pre>

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

```

103 \tabu@newcolumntype \preamble #1{%
104   \ifx\#1\\\tabu@saveerr{}\else
105     \ifundefined{tabu@saved@\string#1}
106       {\tabu@saveerr{#1}}
107       {\csname tabu@saved@\string#1\expandafter\endcsname\expandafter\@ne}%
108   \fi
109 }% \NC@rewrite@\preamble

```

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

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

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

```

110 \tabu@newcolumntype \tabu@rewritefirst{%
111   \tabu@X@cols \z@ \let\NC@rewrite@X \tabu@NC@rewrite@X
112   \let\tabu@dimexpr \relax \aftergroup \tabu@global@temp
113   \aftergroup \tabu@restore@NC@list
114   \NC@list{\NC@do\usetabu \NC@do\preamble \NC@do *}%
115   \tabu@lines % defines NC@rewrite@| for tabu only (inside @mkpream group)
116   %\NC@list\expandafter{\the\NC@list \NC@do X\NC@do\tabu@rewritelast}%
117   %% \message{NC@list est: \the\NC@list^^J}%
118   \NC@list\expandafter{\the\expandafter\NC@list \expandafter\NC@do \expandafter X%
119     \tabu@NC@list \NC@do\tabu@rewritelast}%
120   %% \message{NC@list est: \the\NC@list^^J}%
121   \ifcat$\dollar\end\else
122     \let\save@decl \tabu@save@decl \fi % no inversion of tokens in text mode
123   \NC@find \tabu@rewritelast
124 }% \NC@rewrite@\tabu@rewritefirst
125 \def\tabu@restore@NC@list{\NC@list\expandafter{\tabu@NC@list}}

```

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

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

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

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

```

126 \tabu@newcolumntype \tabu@rewritelast{%
127   \NC@list\expandafter{\tabu@NC@list}%
128   \let\tabu@next \NC@find
129   \ifx \@halignto\relax % found a X column
130     \ifx \tabu@halignto\relax % X column measure
131       \tabu@global@Xnested
132       \iftabu@negcoef \aftergroup\tabu@negcoeftrue \fi
133       \aftergroup \tabu@prep@TRIAL
134     \else \aftergroup \tabu@setup@everycr
135     \fi

```

```

136 \else
137 \aftergroup \tabu@setup@everycr
138 \ifdim \tabu@target=\z@ \else
139 \let\tabu@next \tabu@rewritelast \fi % <need \extracolsep ?>
140 \fi
141 \let\@halignto \tabu@halignto
142 \tabu@next
143 }% \NC@rewrite@\tabu@rewritelast
144 \def\tabu@rewritelast{\futurelet\tabu@temp \tabu@rewrite@last}
145 \def\tabu@rewrite@last{%
146 \ifx \tabu@temp\@sptoken
147 \expandafter\tabu@gobblespace \expandafter\tabu@rewritelast
148 \else
149 {\endlinechar\m@ne\scantokens{\let\x !}}%
150 \expandafter\ifx \x\tabu@temp
151 \def\tabu@next {\NC@find @{\extracolsep\@flushglue}}\else
152 {\endlinechar\m@ne\scantokens{\let\x !}}%
153 \expandafter\ifx \x\tabu@temp
154 \def\tabu@next {\NC@find @{\extracolsep\@flushglue}}\else
155 \def\tabu@next {\NC@find !{\extracolsep\@flushglue}}\fi\fi
156 \expandafter\tabu@next
157 \fi
158 }% \tabu@rewritelast
159 \def\tabu@global@Xnested{\global\expandafter\let
160 \csname tabu@global@X\the\tabu@nested\endcsname\tabu@global@X
161 \aftergroup\let \aftergroup\tabu@global@X
162 \expandafter\aftergroup \csname tabu@global@X\the\tabu@nested\endcsname
163 }% \tabu@global@Xnested

```

`\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.

```
164 \newif\iftabu@colortbl
```

`\tabu@nowrite` A trick (from the T_EX-book to forbidd `\write` when a trial is done on the `\halign`.

```

165 \def\tabu@nowrite#1#{\afterassignment}\toks@}
166 \let\tabu@write\write
167 \let\tabu@immediate\immediate
168 \protected\def\tabu@GenericError{\begingroup
169 \def\immediate\write{\aftergroup\endgroup
170 \tabu@immediate\tabu@write}\tabu@GenericErrorORI}%
171 \let\tabu@GenericErrorORI \GenericError
172 \protected\def\tabu@warn{\begingroup
173 \def\immediate\write{\aftergroup\endgroup
174 \tabu@immediate\tabu@write}\PackageWarning\tabu}

```

8.3 `\tabu`, `\endtabu`, `\longtabu` and `\endlontabu`

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

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

```

175 \def\tabu{%
176 \ifmmode \def\tabu@{\array}\def\endtabu{\endarray}%
177 \else \def\tabu@{\tabular}\def\endtabu{\endtabular}\fi
178 \tabu@setup \tabu@settarget
179 }% \tabu
180 \def\longtabu{%
181 \ifmmode\PackageError\tabu{\longtabu not allowed in math mode}\fi

```

```

182 \def\tabu@{\longtable}\def\endlongtabu{\endlongtable}%
183 \let\tabu@arrayleft@measure \tabu@long@arrayleft@measure
184 \let\tabu@endTRIAL \tabu@long@endTRIAL
185 \tabu@setup \tabu@settarget
186 }% \longtabu
187 \def\tabu@nolongtabu{\PackageError{tabu}
188   {longtabu requires the longtable package}\@ehd}

```

`\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.

```

189 \def\tabu@setup{%
190   \tabu@adl@fix \let\endarray \tabu@endarray % <fix> colortbl & arydshln (delarray)
191   \let\tabu@global@temp \@empty \let\tabu@global@X \@empty
192   \ifx \verbatim\tabu@sanitizetext % <nested tabu>
193     \tabu@setup@saveglobal
194     \advance\tabu@nested \@ne
195     \def\tabu@aligndefault{t}\let\tabu@defaulttarget \linewidth
196   \else
197     \def\tabu@aligndefault{c}\edef\tabu@hfuzz {\the\hfuzz}%
198     \global\tabu@footnotes{}\aftergroup\the\aftergroup\tabu@footnotes
199     \ifdim\parindent>\z@ \ifx\linewidth\tabu@defaulttarget
200       \everypar\expandafter{% % correction for indentation
201         \the\everypar\everypar\expandafter{\the\everypar}%
202         \setbox\z@=\lastbox
203         \ifdim\wd\z@>\z@ \advance\linewidth -\wd\z@\fi
204         \box\z@
205       }%
206     \fi\fi
207   \fi
208   \let\@footnotetext \tabu@footnotetext \let\@xfootnotetext \tabu@xfootnotetext
209   \let\@xfootnote \tabu@xfootnote \let\centering \tabu@centering
210   \let\raggedright \tabu@raggedright \let\raggedleft \tabu@raggedleft
211   \tabu@trivlist %<restore \\\=@centercr inside lists>
212   \def\tabu@decimal {\tabu@tabu@decimal}\let\verbatim \tabu@sanitizetext
213   \let\fbbox \tabu@fbbox \let\rowfont \tabu@rowfont
214   \let\prepnext@tok \tabu@prepnext@tok %<for \rowfont and \tabu@cline>
215   \everycr{}\let\tabu@everycr \everycr
216   \tabu@spreadfalse \tabu@measuringfalse \tabu@negcoeffalse
217   \edef\tabu@NC@list{\the\NC@list}\NC@list{\NC@do\tabu@rewritefirst \NC@do *}%
218 }% \tabu@setup
219 \def\tabu@setup@saveglobal{%
220   \expandafter\xdef\csname tabu@global@\the\tabu@nested\endcsname{%
221     \global\tabu@X@cols \the\tabu@X@cols\relax
222     \global\tabu@naturalX \the\tabu@naturalX\relax
223     \gdef\noexpand\tabu@naturalX@min {\tabu@naturalX@min}%
224     \gdef\noexpand\tabu@naturalX@max {\tabu@naturalX@max}}%
225   \expandafter\aftergroup\csname tabu@global@\the\tabu@nested\endcsname
226 }% \tabu@setup@saveglobal

```

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

```

\tabu@begin
227 \def\tabu@settarget{\futurelet\@let@token \tabu@sett@rget}
228 \def\tabu@sett@rget{\tabu@target\z@
229   \ifcase \ifx \bgroup\@let@token 0\else
230     \ifx [\@let@token 0\else
231     \ifx \@sptoken\@let@token 1\else
232     \if t\@let@token 2\else
233     \if s\@let@token 3\else
234     \m@ne\fi\fi\fi\fi\fi\relax
235   \expandafter\tabu@begin
236   \or \expandafter\tabu@gobblespace\expandafter\tabu@settarget
237   \or \expandafter\tabu@to
238   \or \expandafter\tabu@spread
239   \else \expandafter\tabu@begin
240   \fi
241 }% \tabu@sett@rget
242 \def\tabu@to to{\def\@halignto{to}\tabu@gettarget}
243 \def\tabu@spread spread{\tabu@spreadtrue\def\@halignto{spread}\tabu@gettarget}
244 \def\tabu@gettarget {\afterassignment\tabu@linegoaltarget \tabu@target}
245 \def\tabu@linegoaltarget {\futurelet\tabu@temp \tabu@linegoal@target}
246 \def\tabu@linegoal@target {%
247   \ifx \tabu@temp\LNG\setlinegoal
248     \LNG\setlinegoal \expandafter \@firstoftwo \fi % @gobbles \LNG\setlinegoal
249   \tabu@begin
250 }% \tabu@linegoal@target
251 \def\tabu@begin #1{%
252   \edef\@halignto {\ifdim\tabu@target>\z@ \@halignto\the\tabu@target\fi}%
253   \let\tabu@halignto \@halignto
254   \@testopt \tabu@@begin \tabu@aligndefault #1\@nil
255 }% \tabu@begin
256 \def\tabu@@begin [#1]#2\@nil#3{%
257   \tabu@preparesave{#3}% <for \savetabu>
258   \def\tabu@align {#1}%
259   \tabu@ [{\tabu@align}]#2{\tabu@rewritefirst #3}%
260 }% \tabu@@begin

```

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

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

```

261 \def\tabu@preparesave #1{%
262   \edef\tabu@savedparam{%
263     \ifmmode \col@sep \the\arraycolsep
264     \else \col@sep \the\tabcolsep \fi\relax
265     \arrayrulewidth \the\arrayrulewidth\relax
266     \doublerulesep \the\doublerulesep\relax
267     \extrarowheight \the\extrarowheight\relax
268     \extratabsurround \the\extratabsurround\relax
269     \def\noexpand\arraystretch {\arraystretch}%
270     \iftabu@colortbl \minrowclearance\the\minrowclearance\relax \fi}%
271   \def\tabu@savedpream {\edef\tabu@halignto{to\the\tabu@target}\NC@find #1}%
272 }% \tabu@preparesave

```

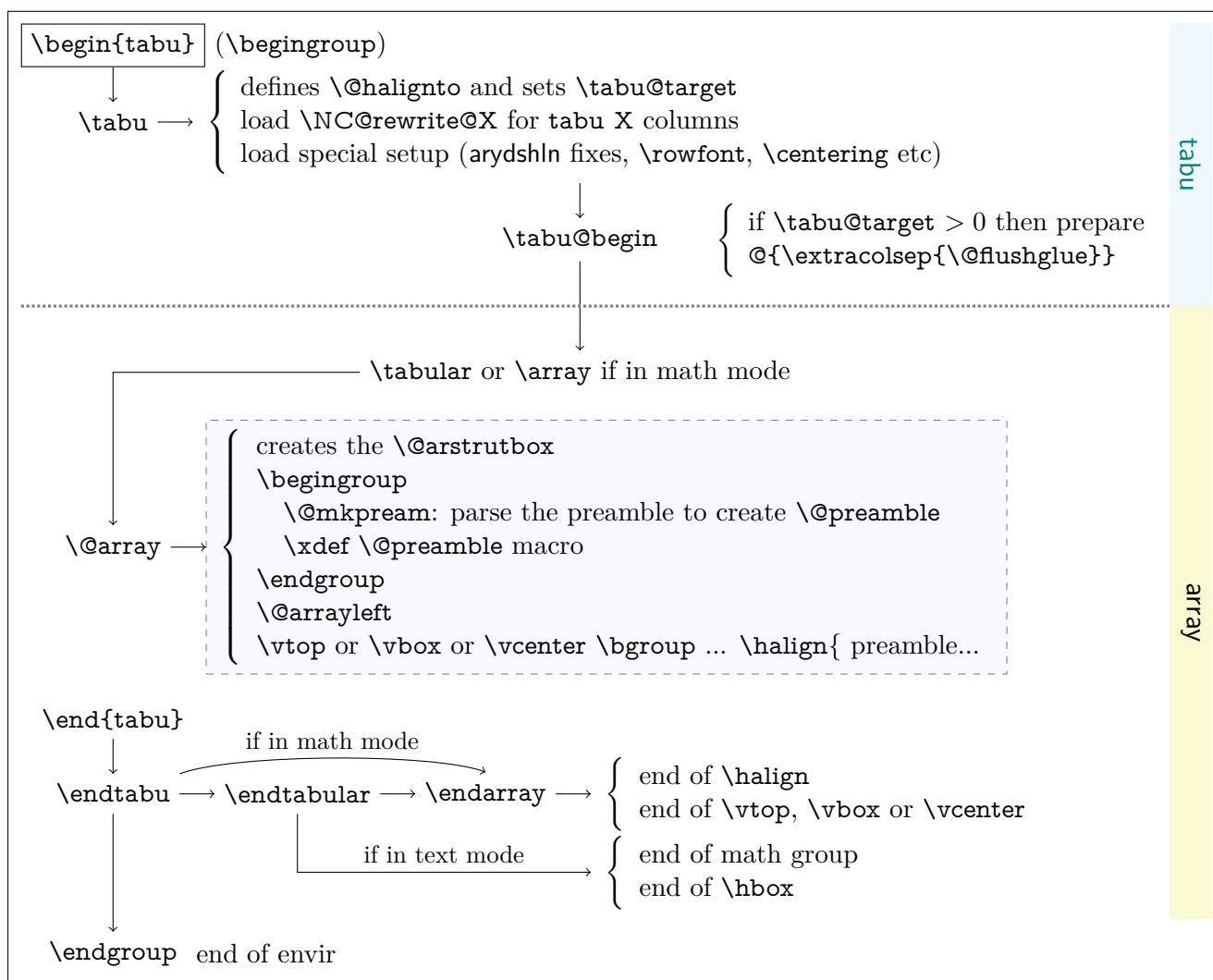
8.4 Footnotes inside tabu

<code>\tabu@footnotes</code>	The (global) token that stores the footnotes inside <code>tabu</code> :
	273 <code>\newtoks\tabu@footnotes</code>
<code>\tabu@footnotetext</code>	The macros in case <code>hyperref</code> is not used, or used with the option <code>hyperfootnotes=false</code> :
	274 <code>\long\def\tabu@footnotetext #1{%</code>
	275 <code>\edef\@tempa{\the\tabu@footnotes</code>
	276 <code>\noexpand\footnotetext [\the\csname c@\mpfn\endcsname]}%</code>
	277 <code>\global\tabu@footnotes\expandafter{\@tempa {#1}}}%</code>
	278 <code>\long\def\tabu@xfootnotetext [#1]#2{%</code>
	279 <code>\global\tabu@footnotes\expandafter{\the\tabu@footnotes</code>
	280 <code>\footnotetext [{#1}]{#2}}%</code>
	281 <code>\let\tabu@xfootnote \@xfootnote</code>
<code>\tabu@Hy@ftntxt</code>	The macros in case <code>hyperref</code> is loaded with the option <code>hyperfootnotes=true</code> :
<code>\tabu@Hy@xfootnote</code>	282 <code>\long\def\tabu@Hy@ftntxt{\tabu@Hy@ftntxt {\the \c@footnote }}</code>
	283 <code>\long\def\tabu@Hy@xfootnote [#1]{%</code>
	284 <code>\begingroup</code>
	285 <code>\value\mpfn #1\relax</code>
	286 <code>\protected@xdef \@thefnmark {\thempfn}%</code>
	287 <code>\endgroup</code>
	288 <code>\@footnotemark \tabu@Hy@ftntxt {#1}%</code>
	289 <code>}% \tabu@Hy@xfootnote</code>
	290 <code>\long\def\tabu@Hy@ftntxt #1#2{%</code>
	291 <code>\edef\@tempa{%</code>
	292 <code>\the\tabu@footnotes</code>
	293 <code>\begingroup</code>
	294 <code>\value\mpfn #1\relax</code>
	295 <code>\noexpand\protected@xdef</code>
	296 <code>\noexpand\@thefnmark{\noexpand\thempfn}%</code>
	297 <code>\expandafter\noexpand\expandafter\tabu@Hy@footnotetext</code>
	298 <code>\expandafter{%</code>
	299 <code>\Hy@footnote@currentHref</code>
	300 <code>}%</code>
	301 <code>}%</code>
	302 <code>\global\tabu@footnotes\expandafter{%</code>
	303 <code>\@tempa {#2}}%</code>
	304 <code>\endgroup</code>
	305 <code>}%</code>
	306 <code>}% \tabu@Hy@ftntxt</code>
	307 <code>\long\def\tabu@Hy@footnotetext #1#2{%</code>
	308 <code>\H@@footnotetext{%</code>
	309 <code>\ifHy@nesting</code>
	310 <code>\hyper@@anchor {#1}{#2}%</code>
	311 <code>\else</code>
	312 <code>\Hy@raisedlink{%</code>
	313 <code>\hyper@@anchor {#1}{\relax}%</code>
	314 <code>}%</code>
	315 <code>\def\@currentHref {#1}%</code>
	316 <code>\let\@currentlabelname \@empty</code>
	317 <code>#2%</code>
	318 <code>\fi</code>
	319 <code>}%</code>
	320 <code>}% \tabu@Hy@footnotetext</code>
	321 <code>\AtBeginDocument{%</code>
	322 <code>\csname\ifcsname ifHy@hyperfootnotes\endcsname</code>
	323 <code>ifHy@hyperfootnotes\else iffalse\fi\endcsname</code>

```
324      \let\tabu@footnotetext \tabu@Hy@ftntext
325      \let\tabu@xfootnote    \tabu@Hy@xfootnote
326  \fi
327 }
```


8.5 Flow chart of expansion

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



`tabu to` with `X` column

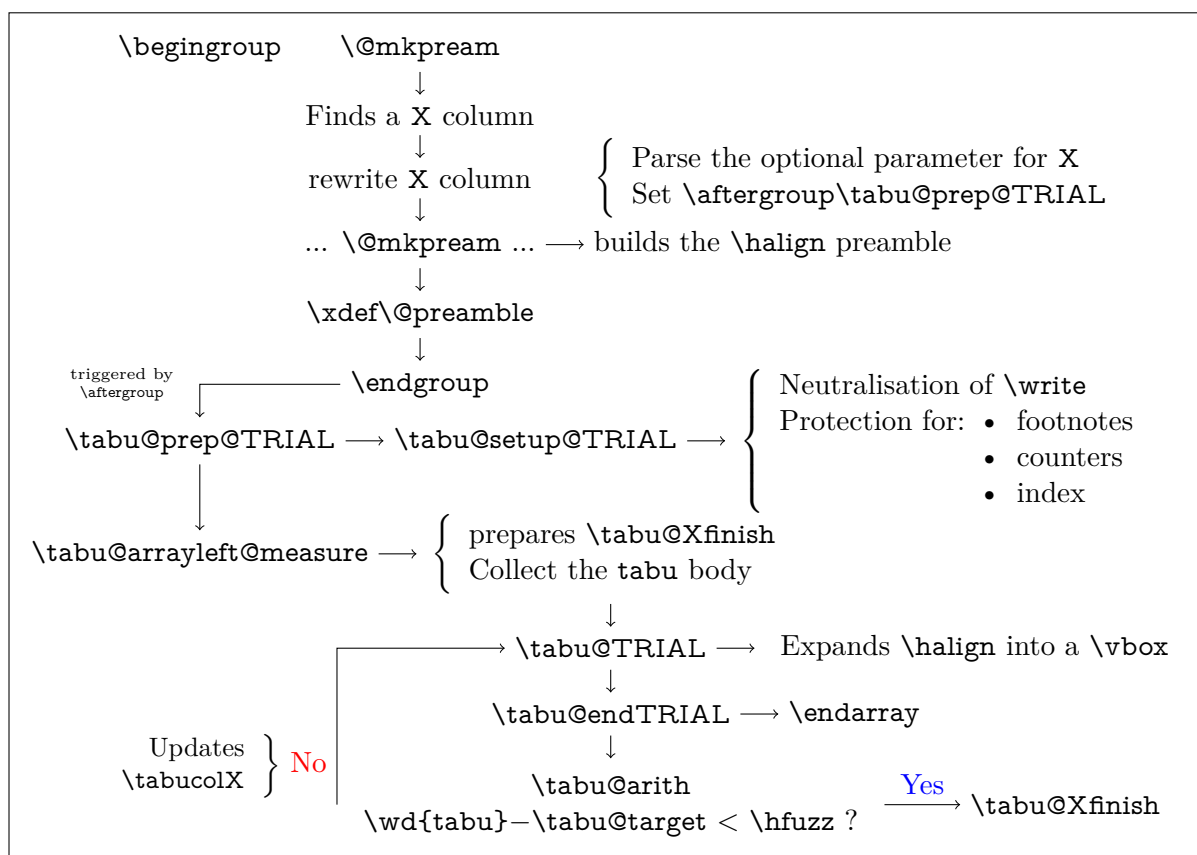
The important part of the job is made inside the dashed box above: `\@mkpreamble` expands the columns definitions, which can be user defined. Hopefully, it does its job inside a group, therefore a user-column can set a macro to be expanded `\aftergroup`. This implementation allows much modifications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

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

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

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

8.6 tabu X column definition

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

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

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

```

328 \def\tabu@rewrite@X #1{\tabu@Xarg {#1}%
329   \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \relax
330   \iftabu@measuring % not the first X column found in preamble
331     \edef\tabu@global@X {\tabu@global@X
332       \advance\tabu@X@cols \@ne
333       \def\csname tabu@X\the\tabu@X@cols\endcsname {\strip@pt\tabu@X@sum}%
334       \advance\tabu@X@sum \the\tabu@X@sum\relax}%
335   \else % first X column found in preamble
336     \tabu@measuringtrue
337     \iftabu@spread \tabu@spreadtarget\tabu@target \tabu@target\z@
338     \else          \tabu@spreadtarget \z@ \fi
339     \ifdim\tabu@target=\z@
340       \setlength\tabu@target \tabu@defaulttarget
341       \tabu@message{(tabu) Default target:
342         \ifx\tabu@defaulttarget\linewidth\string\linewidth
343         \else\ifx\tabu@defaulttarget\linegoal\string\linegoal
344         \fi\fi=\the\tabu@target \on@line, page \the\c@page}\fi
345     \tabu@message{\tabu@titles}%
346     \edef\tabu@global@X {\tabu@global@X
347       \tabu@X@cols \@ne
348       \def\csname tabu@X1\endcsname{\strip@pt\tabu@X@sum}%
349       \tabu@X@sum \the\tabu@X@sum\relax
350       \tabu@target \the\tabu@target\relax
351       \tabu@spreadtarget \the\tabu@spreadtarget\relax}%
352     \let\tabu@halignto \relax
353   \fi
354 }% \tabu@rewrite@X
355 \let\tabu@@@rewrite@X \tabu@rewrite@X

```

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

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

```

356 \def\tabu@rewrite@Xnested #1{\in@ ${#1}\ifin@
357   \def\tabucolX@rw {>{${#1}}p{\tabucolX}<{${#1}}}\else
358   \def\tabucolX@rw {p{\tabucolX}}\fi}
359 \def\tabu@rewrite@Xrestore #1{\tabu@gobbleopt{\def\tabucolX@rw{1}}}

```

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

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

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

```

360 \def\tabu@Xarg#1{%
361   \advance\tabu@X@cols \@ne
362   \tabu@wdlet\tabu@X@cols \z@
363   \ifx\#1\% <shortcut when no option>
364     \tabu@X@sum \p@
365     \edef\tabucolX@rw {p{\tabu@dimexpr{\the\tabu@X@cols}{\tabucolX}}}%
366     \edef\tabu@temp {p{\the\tabucolX}}% <required for \usetabu>
367   \else
368     \tabu@X@sum \z@
369     \let\tabucolX@align \@empty \let\tabucolX@rw \@empty
370     \let\tabu@Xmath \@empty \let\tabu@temp \@empty
371     \tabu@Xparse {}#1\relax\@nnil
372   \fi
373 }% \tabu@Xarg
374 \def\tabu@Xparse #1{\futurelet\@let@token \tabu@Xtest}
375 \expandafter\def\expandafter\tabu@Xparsespace\space{\tabu@Xparse{}}
376 \def\tabu@Xtest{%
377   \ifcase \ifx \@nnil\@let@token \z@ \else
378     \ifx \relax\@let@token \m@ne\else
379     \if ,\@let@token \m@ne\else
380     \if p\@let@token 1\else
381     \if m\@let@token 2\else
382     \if b\@let@token 3\else
383     \if l\@let@token 4\else
384     \if c\@let@token 5\else
385     \if r\@let@token 6\else
386     \if L\@let@token 7\else
387     \if C\@let@token 8\else
388     \if R\@let@token 9\else
389     \ifx \@sptoken\@let@token 10\else
390     \if .\@let@token 11\else
391     \if -\@let@token 11\else
392     \ifcat $\@let@token 12\else
393     13\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\relax
394   \tabucolX@do \expandafter \remove@to@nnil
395   \or \tabu@Xtype {p}%
396   \or \tabu@Xtype {m}%
397   \or \tabu@Xtype {b}%
398   \or \tabu@Xalign \raggedright
399   \or \tabu@Xalign \centering
400   \or \tabu@Xalign \raggedleft
401   \or \tabu@Xalign \RaggedRight
402   \or \tabu@Xalign \Centering
403   \or \tabu@Xalign \RaggedLeft
404   \or \expandafter \tabu@Xparsespace
405   \or \expandafter \tabu@Xcoef
406   \or \let\tabu@Xmath=$\expandafter \tabu@Xparse
407   \or \tabu@Xcoef{}%
408   \else\expandafter \tabu@Xparse
409   \fi
410 }% \tabu@Xtest
411 \def\tabu@Xalign#1{%

```

```

412 \ifx \tabucolX@align\@empty \else \PackageWarning{tabu}
413   {Duplicate horizontal alignment specification}\fi
414 \def\tabucolX@align{#1}\expandafter\tabu@Xparse
415 }% \tabu@Xalign
416 \def\tabu@Xtype#1{%
417   \ifx \tabucolX@rw\@empty \else \PackageWarning{tabu}
418     {Duplicate vertical alignment specification}\fi
419   \def\tabucolX@rw{#1}\expandafter\tabu@Xparse
420 }% \tabu@Xtype
421 \def\tabu@Xcoef#1{\edef\tabu@temp{\tabu@temp#1}%
422   \afterassignment\tabu@Xcoef \tabu@cnt\number\if-#10\fi
423 }% \tabu@Xcoef
424 \def\tabu@Xcoef{%
425   \advance\tabu@X@sum \tabu@temp\the\tabu@cnt\p@
426   \tabu@Xparse}%
427 }% \tabu@Xcoef
428 \def\tabucolX@do{%
429   \ifx\tabucolX@rw\@empty \def\tabucolX@rw{p}\fi
430   \ifdim\tabu@X@sum<\z@ \tabu@negcoeftrue
431     \tabu@X@sum \m@ne\tabu@X@sum
432     \tabu@wdlet \tabu@X@cols \p@
433   \else \ifdim \tabu@X@sum=\z@ \tabu@X@sum \p@ \fi
434   \fi
435   \begingroup
436     \toks@ \expandafter {\tabucolX@align}%
437     \edef\tabu@temp {\strip@pt\tabu@X@sum \tabucolX}%
438     \edef\tabucolX@rw##1{%
439       \ifx$\tabu@Xmath>\{ \noexpand\the\toks@ $\}\else
440       \ifx \tabucolX@align\@empty \else
441         >\{ \noexpand\the\toks@ \}\fi\fi
442       \tabucolX@rw{##1}%
443       \ifx$\tabu@Xmath<\{$\}\fi}%
444     \edef\x{\endgroup
445       \def\noexpand\tabu@temp {\tabucolX@rw{\the\dimexpr\tabu@temp}}%
446       \def\noexpand\tabucolX@rw{%
447         \tabucolX@rw{\tabu@dimexpr{\the\tabu@X@cols}{\tabu@temp}}}%
448     }\x
449 }% \tabucolX@do
450 \def\tabu@dimexpr#1#2{%
451   \ifdim \tabu@wd{#1}>\p@
452     \ifdim \tabu@wd{#1}<#2 \tabu@wd{#1}\else#2\fi\else #2\fi
453 }% \tabu@dimexpr

```

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

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

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

```

454 \def\tabu@prep@TRIAL{%
455   \let\tabu@savedpreamble \@preamble
456   \begingroup \iffalse{\fi \ifnum0='}\fi
457   \tabu@setup@TRIAL
458   \tabucolX \tabu@target
459   \ifcase 0\iftabu@negcoef \else\iftabu@spread \else 1\fi\fi
460   \advance\tabu@X@cols \@ne

```

```

461      \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \@undefined
462      \tabu@X@cols \z@
463      \def\tabu@naturalX@max {\z@}%
464      \def\tabu@naturalX@min {\z@}%
465      \let\tabu@startpboxORI \@startpbox
466      \let\@startpbox \tabu@measure@startpbox
467      \fi
468      \def\tabu@lasttry{\m@ne\p@}\tabu@cnt \z@ % number of trials
469      \tabu@arrayleft@measure
470 }% \tabu@prep@TRIAL

```

`\tabu@setup@TRIAL`

```

471 \def\tabu@setup@TRIAL{\tabu@global@X \let\tabu@global@X \@empty
472   \let\@halignto \@empty \tabu@naturalX \z@
473   \tabu@savecounters \let\tabu@savecounters \relax
474   \hbadness \@M \let\hbadness \@tempcnta
475   \hfuzz \maxdimen \let\hfuzz \@tempdima
476   \let\tabu@rewrite@X \tabu@rewrite@Xnested
477   \let\savetabu \@gobble \let\tabubefaulttarget \linewidth
478   \let\write \tabu@nowrite \let\GenericError \tabu@GenericError
479   \long\def\@xfootnote[##1]##2{}\let\@footnotetext \@gobble
480   \ifx\@arrayright\relax \let\@arrayright \@empty\fi
481   \tabu@TRIAL@hook
482 }% \tabu@setup@TRIAL
483 \def\tabu@savecounters{%
484   \def\@elt##1{\global\value{##1}\the\value{##1}\relax}%
485   \edef\tabu@global@X {\cl@ckpt}\let\@elt \relax
486 }% \tabu@savecounters

```

`\tabuDisableCommands`

```

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

```

8.8 One trial after another

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

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

```

489 \def\tabu@arrayleft@measure#1\bgroup{%
490   \def\tabu@Xfinish{%
491     \tabu@global@Xnested \ifnum0='{}\fi\endgroup
492     \tabu@global@X #1\bgroup}%
493   \def\tabu@TRIAL {\setbox\tabu@box \hbox\bgroup $#1\bgroup
494     \the\toks@ \tabu@endTRIAL}%
495   \toks@{\tabu@setup@everycr
496     \let\@preamble \tabu@savedpreamble}% <required for multicolumn>
497   \tabu@collect
498 }% \tabu@arrayleft@measure

```

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

```

499 \long\def\tabu@long@arrayleft@measure #1\LT@bchunk #2\LT@bchunk{%
500   \def\tabu@Xfinish{%
501     \tabu@global@Xnested \ifnum0='{}\fi\endgroup
502     \tabu@setup@everycr \let\@preamble \tabu@savedpreamble
503     \tabu@global@X #1\LT@bchunk #2\LT@bchunk}%
504   \def\tabu@TRIAL{%
505     \setbox\tabu@box \hbox\bgroup

```



```

506 \tabu@setup@everycr \let\@preamble \tabu@saveditpreamble
507 \tabu@global@X \LTchunksize\@M
508 #1\LT@bchunk #2\LT@bchunk \the\toks@ \tabu@endTRIAL}%
509 \toks@{}%\let\LT@output\relax
510 \tabu@collect
511 }%\tabu@long@arrayleft@measure
512 \def\tabu@long@endTRIAL{%
513 \LT@echunk
514 \global\setbox\@ne \hbox{\unhbox\@ne}\kern\wd\@ne
515 \LT@get@widths
516 \egroup % got \tabu@box
517 \iftabu@spread \tabu@spreadarith % <only once>
518 \else \tabu@arith
519 \fi
520 \iftabu@measuring \tabu@measuringfalse
521 \expandafter \tabu@TRIAL % <continue trials>
522 \else
523 \expandafter \tabu@Xfinish
524 \the \expandafter\toks@ \tabu@endenvir % <we are then!>
525 \fi
526 }%\tabu@long@endTRIAL

```

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

```

527 \def\tabu@collect{\catcode'\^^@=13\def\tabu@stack{b}\tabu@collectbody}
528 \long\def\tabu@collectbody#1\end#2{%
529 \edef\tabu@stack{\tabu@pushbegins #1\begin\end\expandafter\@gobble\tabu@stack}%
530 \ifx\tabu@stack\@empty
531 \toks@\expandafter{\the\toks@#1}\def\tabu@endenvir{\end{#2}}%
532 \expandafter \tabu@TRIAL
533 \else
534 \toks@\expandafter{\the\toks@#1\end{#2}}%
535 \expandafter \tabu@collectbody
536 \fi
537 }%\tabu@collectbody
538 \long\def\tabu@pushbegins#1\begin#2{%
539 \ifx\end#2\else b\expandafter\tabu@pushbegins\fi}

```

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

```

540 \def\tabu@endTRIAL{%
541 \tabu@adl@endTRIAL % <arydshln in nested trials>
542 \endarray$\egroup
543 \iftabu@spread \tabu@spreadarith % <only once>
544 \else \tabu@arith
545 \fi
546 \iftabu@measuring \tabu@measuringfalse
547 \expandafter \tabu@TRIAL % <continue trials>
548 \else
549 \expandafter \tabu@Xfinish
550 \the \expandafter\toks@ \tabu@endenvir % <we are then!>
551 \fi
552 }%\tabu@endTRIAL
553 \def\tabu@adl@endTRIAL{\crrc \noalign{\global\adl@ncol \tabu@nbc}}

```

8.9 Algorithms: Measuring the tabu

The arithmetic of X columns

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

```

554 \def\tabu@arith{%
555   \advance\tabu@cnt \@ne \tabu@message{\tabu@message@arith}%
556   \ifx \@startpbox\tabu@measure@startpbox
557     \let\@startpbox \tabu@startpboxORI \fi
558   \dimen@ \tabu@X@sum
559   \iftabu@negcoef \tabu@arith@negcoef \fi
560   \ifcase 0\ifdim \dimexpr\wd\tabu@box-\tabu@hfuzz-1sp<\tabu@target
561     \else \ifdim \tabu@lasttry=\wd\tabu@box 1\else
562       2\fi\fi\relax
563     \ifdim \tabu@X@sum=\dimen@ \tabu@message{\tabu@message@reached}%
564     \else\ifdim\tabu@X@sum=\z@ \tabu@message{\tabu@message@reached}%
565     \else\ifnum \tabu@cnt>\@ne \tabu@message{\tabu@message@reached}%
566     \else \tabu@measuringtrue
567       \tabucolX = \dimexpr\tabucolX*\p@/\tabu@X@sum\relax
568     \fi\fi\fi
569   \or \tabu@message{\tabu@header Reached minimum width. Backing up}%
570     \tabu@warn{\tabu width = \the\wd\tabu@box\space>
571       \the\tabu@target\space+ \tabu@hfuzz}%
572   \or
573     \ifdim \tabu@X@sum=\z@ \tabu@measuringtrue \tabucolX=\@tempdima
574     \else \tabu@message{\tabu@message@arith}%
575       \advance\tabucolX-\dimexpr(\wd\tabu@box-\tabu@target-\tabu@X@sum/\p@)
576       *\p@/\tabu@X@sum\relax
577     \ifdim \tabucolX<\z@
578       \tabu@warn{X Columns too narrow (table too wide)}%
579       \tabucolX=\tabucolX@error
580     \else \tabu@measuringtrue
581       \ifdim \tabu@X@sum=\dimen@
582         \edef\tabu@lasttry{\the\wd\tabu@box}%
583       \else
584         \ifnum \tabu@cnt>\@ne \edef\tabu@lasttry{\the\wd\tabu@box}\fi
585         \ifdim \tabucolX<\@tempdima \tabucolX \@tempdima\fi
586       \fi
587     \fi
588   \fi
589   \fi %(ifcase)
590   \iftabu@measuring
591   \else \edef\tabu@global@X{\tabu@global@X
592     \tabucolX \the\tabucolX\relax
593     \tabu@target \the\tabu@target\relax % < for \savetabu >
594     \ifnum \tabu@nested=\z@
595       \ifx \@arrayright\@empty
596         \ifdim \tabu@X@sum>\z@
597           \def\noexpand\@halignto{to\the\tabu@target}\fi\fi\fi}%
598     \fi
599     \tabu@X@sum \dimen@
600 }% \tabu@arith
601 \def\tabu@arith@negcoef{%
602   \tabu@X@cols \@ne \count@ \tabu@X@cols
603   \@whiles \ifcsname tabu@X\the\tabu@X@cols\endcsname\fi{%
604     \ifdim \tabu@wd\tabu@X@cols>\p@
605       \expandafter\let\expandafter\tabu@temp
606       \csname tabu@X\the\tabu@X@cols\endcsname

```



```

657 \expandafter\tabu@msg@align \the\wd\tabu@box {}{}{}{}{}\@@
658 <\expandafter\tabu@msg@align \the\tabu@target { }{ }{ }{ }\@@
659 + \tabu@hfuzz\space= hfuzz *****~J%
660 (tabu)\tabu@spaces***** Reached Target:
661     hfuzz = \tabu@hfuzz\on@line\space *****~J
662     \ifnum\tabu@nested=\z0~J\fi}%
663 \def\tabu@message@negcoef #1{\tabu@spaces\tabu@spaces
664 * #1. X[-\csname tabu@X#1\endcsname]: natural width
665 = \tabu@wd{#1}
666 < \csname tabu@X#1\endcsname X
667 = \the\dimexpr\csname tabu@X#1\endcsname\tabucolX
668 }% \tabu@message@negcoef
669 \edef\tabu@spaces{\@spaces}
670 \def\do#1{%
671   \def\tabu@msg@align##1.##2##3##4##5##6##7##8##9\@@{%
672     \ifnum##1<10 #1 #1\else
673     \ifnum##1<100 #1 \else
674     \ifnum##1<\@m #1\fi\fi\fi
675     ##1.##2##3##4##5##6##7##8#1}%
676   \def\tabu@header{(tabu) \ifnum\tabu@cnt<10 #1\fi\the\tabu@cnt) #1}%
677   \def\tabu@titles{\ifnum \tabu@nested=\z0
678     (tabu) Try#1 #1 tabu X #1 #1 #1tabu Width #1 #1 Target
679       #1 #1 #1 #1 Coefs #1 #1 #1 Update~J\fi}%
680   \def\tabu@spreadheader{%
681     (tabu) Try#1 #1 Spread #1 #1 tabu Width #1 #1 Nat. Min. #1 #1 #1 Nat.X
682       #1 #1 #1 New Target~J%
683     (tabu)spread}%
684   \def\tabu@tracing@save##1{\begingroup
685     \def\colsep #####1\relax {(tabu) column sep #1 #1 #1 = #####1~J}%
686     \def\arrayrulewidth #####1\relax{(tabu) arrayrulewidth #1 = #####1~J}%
687     \def\doublerulesep #####1\relax {(tabu) doublerulesep #1 #1= #####1~J}%
688     \def\extrarowheight #####1\relax{(tabu) extrarowheight #1 = #####1~J}%
689     \def\extratabsurround #####1\relax{(tabu) extratabsurround = #####1~J}%
690     \def\minrowclearance #####1\relax {(tabu) minrowclearance#1 = #####1~J}%
691     \def\arraystretch #####1{(tabu) arraystretch #1 #1 = #####1~J}%
692     \def\NC@find #####1\@nil{(tabu) tabu preamble#1 #1 = \detokenize{#####1}~J}%
693     \def\tabu@wdddef#####1####2{(tabu) Natural width #####1 = ####2~J}%
694     \let\edef \@gobbletwo \let\def \@empty
695     \tabu@message{%
696       (tabu) \string\savetabu{##1}: \on@line~J%
697     \ifx \savetabu\tabusaveX
698       (tabu) X columns width #1 = \the\tabucolX~J\fi
699       (tabu) target #1 #1 #1 #1 #1 = \the\tabu@target~J%
700       \tabu@sav@param
701       \tabu@sav@pream \@nil
702       \iftabu@negcoef \tabu@savewd\fi}%
703     \endgroup}%
704 } \do{ }

```

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

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

Nested trials may occur (`tabu spread` inside a `X` column with negativ coefficient for example). For the furthur trials, the standard scheme for `X` column is used: the natural width is measured only once.

pdfT_EX font expansion is disabled inside the `varwidth` environment (we set `\pdfadjustspacing` to 0).

```

705 \def\tabu@measure@startpbox#1{\bgroup
706   \let\@startpbox \tabu@startpboxORI           % in case of nesting
707   \in@{\tabu@dimexpr}{#1}%
708   \ifin@
709     \global\advance\tabu@X@cols \@ne
710     \ifcsname tabu@X\the\tabu@X@cols\endcsname\else
711       \global\tabu@X@cols \@ne
712       \global\tabu@naturalX \z@
713     \fi
714   \fi
715   \iftabu@spread\else
716     \ifin@ {\let\tabu@dimexpr \@firstoftwo\expandafter}%
717       \ifdim \tabu@wd{#1}<\p@ \in@false\fi\fi\fi
718     \ifin@
719       \begin{group} \let\tabu@dimexpr \@secondoftwo
720         \edef\x{\endgroup \def\noexpand\tabu@temp{\the\dimexpr#1}%
721       }\x
722       \setbox\tabu@box=\hbox\bgroup
723       \let\tabu@rewrite@X \tabu@@rewrite@X
724       \begin{varwidth}{\tabu@temp}\narrowragged\arraybackslash
725         \tabu@noadjustspacing \parfillskip\@flushglue
726       \bgroup \aftergroup\tabu@measure@endpbox
727     \else
728       \expandafter\@gobble\@startpbox{#1}% \@gobble \bgroup
729     \fi
730 }% \tabu@measure@startpbox
731 \ifdefined\pdfadjustspacing
732   \def\tabu@noadjustspacing{\pdfadjustspacing\z@}%
733 \else
734   \let\tabu@noadjustspacing\relax
735 \fi
736 \def\tabu@measure@endpbox{%
737   \@finalstrut\@arstrutbox\end{varwidth}\egroup      % <got my \tabu@box>
738   \global\advance\tabu@naturalX \wd\tabu@box
739   \ifdim \tabu@naturalX@max<\tabu@naturalX
740     \xdef\tabu@naturalX@max{\the\tabu@naturalX}\fi
741   \@tempdima=\dimexpr \wd\tabu@box * \tabu@X@sum /
742     \dimexpr\csname tabu@X\the\tabu@X@cols\endcsname\p@\relax\relax
743   \ifdim \tabu@naturalX@min<\@tempdima
744     \xdef\tabu@naturalX@min{\the\@tempdima}\fi
745   \ifdim \tabu@wd\tabu@X@cols>\z@
746   \ifdim \tabu@wd\tabu@X@cols<\wd\tabu@box
747     \tabu@wddef\tabu@X@cols {\the\wd\tabu@box}%
748     \tabu@debug{\tabu@spaces\tabu@spaces
749       * \the\tabu@X@cols. X[-\csname tabu@X\the\tabu@X@cols\endcsname]:
750       target=\the\dimexpr\tabu@temp\relax\space
751       \expandafter\expandafter\expandafter\string\tabu@wd\tabu@X@cols
752       =\tabu@wd\tabu@X@cols}%

```

```

753 \fi\fi
754 \box\tabu@box \egroup % end of \vtop (measure)
755 }% \tabu@measure@endpbox
756 \def\tabu@wd#1{\csname tabu@W\the\tabu@nested @\number1\endcsname}
757 \def\tabu@wddef#1{\expandafter\xdef
758 \csname tabu@W\the\tabu@nested @\number1\endcsname}
759 \def\tabu@wdlet#1{\global\expandafter\let
760 \csname tabu@W\the\tabu@nested @\number1\endcsname}
761 \def\tabu@naturalX@min{\z@}
762 \def\tabu@naturalX@max{\z@}

```

8.10 \everycr setup and \tabuphantomline

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

```

763 \def\tabu@setup@everycr{%
764 \tabu@firstcline true
765 \let\tabu@first@cline \z@
766 \def\tabu@restoreeverycr {\let\everycr \tabu@everycr}%
767 \def\everycr{\afterassignment\tabu@restoreeverycr \@temptokena}% <for ialign>
768 \tabu@everycr\expandafter{\the\tabu@everycr
769 \noalign{%
770 \ifcase\tabu@first@cline
771 \global\let\tabu@first@cline \tw@
772 \or\global\let\tabu@first@cline \tw@
773 \or\global\tabu@firstcline false
774 \global\let\tabu@first@cline \m@ne
775 \fi}%
776 \tabu@rowfontreset
777 }%
778 }% \tabu@setup@everycr

```

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

```

779 \def\tabuphantomline{\crrc\noalign{%
780 \global\let\@arstrutbox \voidb@x
781 \iftabu@colortbl
782 \xdef\tabu@minrowclearance{\the\minrowclearance}%
783 \global\minrowclearance \z@
784 \fi
785 \global\let\vcenter \vbox
786 \ifnum \tabu@nested>\z@ \vspace*{\dimexpr\extrarowheight-\ht\@arstrutbox}\fi
787 \toks@{}\count@\@ne
788 \@whilenum\count@<\tabu@nbc\do{\advance\count@\@ne
789 \toks@\expandafter{\the\toks@&}}%
790 \toks@\expandafter{\the\toks@
791 \cr\noalign{%
792 \global\let\@arstrutbox \tabu@arstrutbox
793 \global\let\vcenter \tabu@vcenter
794 \iftabu@colortbl
795 \global\minrowclearance \tabu@minrowclearance\relax\fi
796 \global\let\tabu@first@cline \@ne}}%
797 \expandafter}\the\toks@
798 }% \tabuphantomline

```


8.11 Lines inside tabu

Vertical lines

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

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

```

799 \def\tabu@lines{{\endlinechar\m@ne \scantokens{\def\x{}}}%
800   \expandafter}\expandafter\tabu@lines@ \x
801 }% \tabu@lines
802 \def\tabu@lines@#1{%
803   \tabu@newcolumnntype #1{\tabuvline@rewrite}%
804   \NC@list\expandafter{\the\NC@list \NC@do #1}%
805 }% \tabu@lines@
806 \newcommand*\tabuvline@rewrite[1][\]{\tabuvline@arg{#1}%
807   \expandafter \NC@find \tabu@temp}
808 \def\tabu@vline#1{\vrule width#1}
```

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

```

809 \def\tabuvline@arg#1{%
810   \ifx\#1\% <shortcut when no option>
811     \edef\tabu@temp{!}%
812     \iftabu@colortbl\beginngroup\noexpand\CT@arc@ \fi
813     \noexpand\tabu@vline\arrayrulewidth
814     \iftabu@colortbl\endgroup\fi}%
815   \else \futurelet \tabu@temp \tabuvline@argi #1\p@\p@\@nnil \tabuvline@argiii
816   \fi
817 }% \tabuvline@arg
818 \def\tabuvline@argi{%
819   \let\tabu@color \@empty
820   \ifcat A\noexpand\tabu@temp
821     \@tempdima\arrayrulewidth
822     \expandafter\tabu@getlinecolor
823   \else \expandafter\tabuvline@argii
824   \fi
825 }% \tabuvline@argi
826 \def\tabuvline@argii#1\@nnil{%
827   \tabu@maybecolor \@tempdima #1 \@nnil
828 }% \tabuvline@argii
829 \def\tabuvline@argiii{%
830   \edef\tabu@temp{!\beginngroup
831     \ifx\tabu@color\@empty
832       \iftabu@colortbl \noexpand\CT@arc@ \fi
833     \else \tabu@color\fi
834     \noexpand\tabu@vline{\the\@tempdima}\endgroup}}%
835 }% \tabuvline@argiii
```

Horizontal lines: `\tabucline`

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

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

```

836 \def\tabu@linedash{4pt}% <default value>
837 \def\tabu@dashgap{4pt}% <default value>
838 \def\tabucline{\noalign{\ifnum0='}\fi \@ifstar
```



```

839 {\@tempswatrue\tabu@ccline}
840 {\@tempswafalse\tabu@ccline}}%
841 \newcommand*\tabu@ccline[2][\toks@{}%
842 \tabu@startstop{#2}\tabu@getlinespec{#1}\@multicnt\@ne
843 \ifnum\tabu@start>\tabu@stop
844 \else\ifx\tabu@xleaders\relax
845 \else
846 \toks@{\noalign{\global\let\@arstrutbox \voidb@x}}%
847 %% \iftabu@firstccline\if\tabu@align t\vskip-\ht\@arstrutbox\fi\fi
848 \@whilenum\@multicnt<\tabu@start\do{\advance\@multicnt\@ne
849 \toks@\expandafter{\the\toks@ &}}%
850 \loop
851 \toks@\expandafter{\the\toks@ \omit }%
852 \ifnum \@multicnt=\@ne
853 \expandafter \tabu@add \csname tabu@tok@\the\@multicnt L\endcsname \@nil
854 \fi
855 \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
856 \toks@\expandafter{\the\toks@ \begingroup}%
857 \toks@\expandafter{\the\expandafter\toks@
858 \tabu@xleaders \endgroup}%
859 \fi
860 \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
861 \expandafter \tabu@add \csname tabu@tok@\the\@multicnt R\endcsname \@nil
862 \fi
863 \ifnum\@multicnt<\tabu@stop
864 \advance\@multicnt\@ne
865 \if@tempswa \toks@\expandafter{\the\toks@ &}%
866 \else \toks@\expandafter{\the\toks@ \span}\fi
867 \repeat
868 \@whilenum\@multicnt<\tabu@nbccls\do{\advance\@multicnt\@ne
869 \toks@\expandafter{\the\toks@ &}}%
870 \toks@\expandafter{\the\toks@ \cr
871 \noalign{\global\let\@arstrutbox \tabu@arstrutbox}}%
872 %% \tabu@debug{LIGNE AJOUTEE:\the\toks@}%
873 \fi\fi
874 \ifnum0='{ \fi\expandafter}\the\toks@
875 }% \tabu@ccline
876 \def\tabu@add #1\@nil{\toks@\expandafter{\the\toks@
877 \begingroup #1\endgroup\ifmmode$\fi}}

```

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

```

878 \def\tabu@startstop#1{\tabu@start@stop #1\relax 1-\tabu@nbccls\@nnil}
879 \def\tabu@start@stop #1-#2\@nnil{%
880 \@defaultunits \tabu@start \number0#1\relax\@nnil
881 \@defaultunits \tabu@stop \number0#2\relax\@nnil
882 \ifnum\tabu@start>\tabu@nbccls\tabu@start=\tabu@nbccls\else
883 \ifnum\tabu@start>\z@\else \tabu@start=\@ne \fi\fi
884 \ifnum\tabu@stop>\tabu@nbccls \tabu@stop\tabu@nbccls \else
885 \ifnum\tabu@stop>\z@\else \tabu@stop=\tabu@nbccls \fi\fi
886 }% \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).

```

887 \def\tabu@getlinespec#1{\let\tabu@xleaders \relax
888   \@defaultunits \let\@tempa=#1 \relax\@nnil
889   \ifx\@tempa\relax \let\tabu@xleaders \tabu@defaultleaders\else
890   \ifx\@tempa\hbox \tabu@defleaders{#1}\else
891   \ifx\@tempa\box \tabu@defleaders{#1}\else
892   \ifx\@tempa\copy \tabu@defleaders{#1}\else
893   \ifcsname tabu@line@style@\detokenize{#1}\endcsname
894     \csname tabu@line@style@\detokenize{#1}\endcsname
895   \else \tabu@getline{#1}\p@ on0pt off0pt\fi\fi\fi\fi\fi
896 }% \tabu@getlinespec
897 \def\tabu@defleaders#1{%
898   \def\tabu@xleaders{\iftabu@colortbl\CT@arc@\fi
899     \xleaders\hbox{\lower.5\extrarowheight#1}\tabu@leaderfill}}
900 \def\tabucline@warn#1{\PackageWarning{tabu}
901   {Undefined line syle: #1
902   \MessageBreak Using default line style instead}}%
903   \let\tabu@xleaders \tabu@defaultleaders
904 }% \tabucline@warn

```

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

```

905 \def\tabu@getline#1{\tabu@lineon #1 \@nil on\tabu@linedash \p@ \@nil\@nnil{#1}}
906 \def\tabu@maybecolor{\ifx\tabu@color\@empty \afterassignment\tabu@colortest
907   \else \@defaultunits\fi}
908 \def\tabu@lineon #1on#2\@nil#3\@nnil#4{%
909   \let\tabu@color \@empty
910   \@defaultunits \let\@tempa=#1 \relax\@nnil
911   \ifcase 0\ifx o\@tempa 1\else\ifcat A\noexpand\@tempa 2\fi\fi\relax
912     \tabu@maybecolor \@tempdima #1\arrayrulewidth \p@\@nnil
913     \tabu@maybecolor \@tempdimb #2\p@ \@nnil
914     \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil
915   \or\@tempdima \arrayrulewidth
916     \tabu@maybecolor \@tempdimb #2\p@ \@nnil
917     \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil
918   \else
919     \tabu@maybecolor \@tempdima \arrayrulewidth #1\p@\@nnil
920     \ifx\tabu@color\@empty \tabucline@warn\@tempa\else
921     \@tempdimb \z@
922     \tabu@lineoff off0pt \p@\@nnil \fi
923   \fi
924 }% \tabu@lineon
925 \def\tabu@lineoff #1off#2\@nnil{%
926   \tabu@maybecolor \@tempdimc #2\p@ \@nnil
927   \iftabu@colortbl \ifx \tabu@color\@empty
928     \def\tabu@color{\noexpand\CT@arc@}%
929   \fi\fi
930   \ifdim \@tempdimb=\z@
931     \ifdim \@tempdimc>\z@ \@tempdimb \tabu@dashgap\relax\fi\fi
932   \ifdim \@tempdimc=\z@
933     \ifdim \@tempdimb>\z@ \@tempdimc \tabu@linedash\relax\fi\fi
934   \ifdim \@tempdima<\z@ \else
935   \ifdim \@tempdimb<\z@ \else
936   \ifdim \@tempdimc<\z@ \else
937     \edef \tabu@xleaders{\tabu@color\xleaders
938       \ifdim \@tempdimc>\z@

```

```

939         \hbox\bgroup \kern\the\dimexpr\@tempdimc/2\relax\fi
940         \noexpand\iftabu@firstcline
941         \vrule depth\dimexpr\the\@tempdima
942             \ifdim\@tempdimb>\z@ width\the\@tempdimb\fi
943         \noexpand\else
944         \vrule height\dimexpr-\extrarowheight+\the\@tempdima
945             depth\dimexpr\extrarowheight
946             \ifdim\@tempdimb>\z@ width\the\@tempdimb\fi
947         \noexpand\fi
948         \ifdim\@tempdimc>\z@
949             \kern\the\dimexpr\@tempdimc/2\egroup\fi
950         \tabu@leaderfill}%
951     \fi\fi\fi
952 }% \tabu@lineoff

```

`\tabu@colortest`

```

953 \def\tabu@colortest{\futurelet\tabu@temp\tabu@linecolor}
954 \def\tabu@linecolor{%
955     \ifcase 0\if ,\noexpand\tabu@temp\else
956         \ifx\relax\tabu@temp\else
957         \ifx \@sptoken\tabu@temp1\else
958         \ifcat A\noexpand\tabu@temp2\else
959             3\fi\fi\fi\fi\relax
960     \def\tabu@next##1{\futurelet\tabu@temp\tabu@linecolor}%
961     \or\def\tabu@next{\tabu@gobblespace{\futurelet\tabu@temp\tabu@linecolor}}%
962     \or\let\tabu@next \tabu@getlinecolor
963     \else\expandafter\remove@to@nnil
964     \fi \tabu@next
965 }% \tabu@linecolor
966 \def\tabu@getlinecolor#1\p@{%
967     \edef\@tempa{\zap@space #1 \@empty}%
968     \ifcsname\string\color@\@tempa\endcsname
969         \edef\tabu@color{\noexpand\noexpand\noexpand\color{\@tempa}}% \set@color
970     \fi\remove@to@nnil
971 }% \tabu@getlinecolor

```

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

```

972 \def\tabulinestyle#1{\@for\@tempa:=#1\do
973     {\expandafter\tabu@linestyle\@tempa==\@nil}}
974 \def\tabu@linestyle#1=#2=#3\@nil{%
975     \begingroup \tabu@getlinespec {#2}\expandafter\gdef
976         \csname tabu@line@style@\detokenize{#1}\expandafter\endcsname
977         \expandafter{\expandafter\def\expandafter\tabu@xleaders
978             \expandafter{\tabu@xleaders}}}%
979     \endgroup
980 }% \tabu@linestyle
981 \expandafter\def \csname tabu@line@style@\endcsname {%
982     \let\tabu@xleaders \tabu@defaultleaders}%
983 \def\tabu@defaultleaders{\iftabu@colortbl\CT@arc@\fi
984     \leaders
985     \iftabu@firstcline
986         \vrule depth \arrayrulewidth
987     \else
988         \vrule height\dimexpr-\extrarowheight+\arrayrulewidth
989         depth \extrarowheight
990     \fi
991     \tabu@leaderfill}
992 \let\tabu@leaderfill \hfil

```

8.12 Verbatim inside tabu with X columns

`\tabu@sanitizetext`

```

993 {\catcode32=13\relax\catcode'\^@=13\relax
994 \gdef\tabu@verb{\@sanitize\catcode'\^=7\edef\~{\string~}\tabu@makeatletter
995 \catcode32=13\let ~=\ \catcode'\^@=13\def~@{\hskip\parfillskip\null\linebreak}%
996 \endlinechar\m@ne}}
997 {\catcode64=13\edef @{\string @}\expandafter\xdef
998   \csname tabu@makeatletter\endcsname{\catcode64=13\def\noexpand @{\unskip @}}
999 \newcommand\tabu@sanitizetext[1][\ttfamily]{\begingroup
1000   \tabu@verb #1\tabu@s@nitizetext}
1001 \long\def\tabu@s@nitizetext#1{\@makeother\{\@makeother\}%
1002   \everyeof{\noexpand}\scantokens{#1}\endgroup}

```

8.13 Numbers in tabu

`\tabudecimal`

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

```

1003 \def\tabu@tabudecimal#1{%
1004   \def\tabu@decimal{#1}\@temptokena{}%
1005   \let\tabu@getdecimal@ \tabu@getdecimal@ignorespaces
1006   \tabu@scandecimal
1007 }% \tabu@tabudecimal
1008 \def\tabu@scandecimal{\futurelet \tabu@temp \tabu@getdecimal@}
1009 \def\tabu@skipdecimal#1{#1\tabu@scandecimal}
1010 \def\tabu@getdecimal@ignorespaces{%
1011   \ifcase 0\ifx\tabu@temp\ignorespaces\else
1012     \ifx\tabu@temp\@sptoken1\else
1013       2\fi\fi\relax
1014     \let\tabu@getdecimal@ \tabu@getdecimal
1015     \expandafter\tabu@skipdecimal
1016   \or \expandafter\tabu@gobblespace\expandafter\tabu@scandecimal
1017   \else \expandafter\tabu@skipdecimal
1018   \fi
1019 }% \tabu@getdecimal@ignorespaces
1020 \def\tabu@get@decimal#1{\@temptokena\expandafter{\the\@temptokena #1}%
1021   \tabu@scandecimal}
1022 \def\do#1{%
1023   \def\tabu@get@decimalspace#1{%
1024     \@temptokena\expandafter{\the\@temptokena #1}\tabu@scandecimal}%
1025 }\do{ }
1026 \let\tabu@@tabudecimal \tabu@tabudecimal

```

`\tabu@getdecimal`

```

1027 \def\tabu@getdecimal{%
1028   \ifcase 0\ifx 0\tabu@temp\else
1029     \ifx 1\tabu@temp\else
1030     \ifx 2\tabu@temp\else
1031     \ifx 3\tabu@temp\else
1032     \ifx 4\tabu@temp\else
1033     \ifx 5\tabu@temp\else
1034     \ifx 6\tabu@temp\else
1035     \ifx 7\tabu@temp\else
1036     \ifx 8\tabu@temp\else
1037     \ifx 9\tabu@temp\else
1038     \ifx .\tabu@temp\else
1039     \ifx ,\tabu@temp\else

```

```

1040          \ifx -\tabu@temp\else
1041          \ifx +\tabu@temp\else
1042          \ifx e\tabu@temp\else
1043          \ifx E\tabu@temp\else
1044          \ifx\tabu@celleft\tabu@temp1\else
1045          \ifx\ignorespaces\tabu@temp1\else
1046          \ifx\@sptoken\tabu@temp2\else
1047          3\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\relax
1048      \expandafter\tabu@get@decimal
1049  \or \expandafter\tabu@skipdecimal
1050  \or \expandafter\tabu@get@decimalspace
1051  \else\expandafter\tabu@printdecimal
1052  \fi
1053 }% \tabu@getdecimal
1054 \def\tabu@printdecimal{%
1055   \edef\tabu@temp{\the\@temptokena}%
1056   \ifx\tabu@temp\@empty\else
1057   \ifx\tabu@temp\space\else
1058     \expandafter\tabu@decimal\expandafter{\the\@temptokena}%
1059   \fi\fi
1060 }% \tabu@printdecimal

```

8.14 \savetabu

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

```

1061 \newcommand*\savetabu[1]{\noalign{%
1062   \ifx\#1\\ \tabu@savewarn{}\The tabu will not be saved\else
1063     \ifundefined\tabu@saved@\string#1\{\}\{\tabu@savewarn{#1}\Overwritting}\}%
1064     {%
1065     \toks0\expandafter{\@preamble}%
1066     \toks1\expandafter{\tabu@savedparam}%
1067     \toks2\expandafter{\tabu@savedpream}%
1068     \iftabu@colortbl
1069       \toks3\expandafter{\expandafter\gdef\expandafter\CT@arc@
1070         \expandafter{\CT@arc@}}%
1071       \toks3\expandafter{\the\toks\expandafter3
1072         \expandafter\gdef\expandafter\CT@drsc@\expandafter{\CT@drsc@}}%
1073     \else \toks3{}\%
1074     \fi
1075     \iftabu@negcoef
1076       \edef\tabu@savewd{\tabu@savewd{\number\numexpr1}}%
1077       \toks3\expandafter{\the\toks\expandafter3\tabu@savewd}\fi
1078     \expandafter\xdef\csname\tabu@saved@\string#1\endcsname##1{%
1079       \noexpand\ifcase##1\relax
1080         \noexpand\ifdim\tabu@target>\z@
1081           \noexpand\tabu@warn@usetabu
1082         \noexpand\fi
1083         \tabu@target \the\tabu@target\relax
1084         \tabucolX \the\tabucolX\relax
1085         \the\toks1%
1086         \gdef\noexpand\tabu@usetabu{%
1087           \the\toks1%
1088           \def\noexpand\@halignto{\@halignto}%
1089           \tabucolX \the\tabucolX\relax
1090           \def\noexpand\@preamble{\the\toks0}%
1091           \noexpand\tabu@make@arstrutbox
1092           \the\toks3

```

```

1093             \noexpand\ifx \noexpand\tabu@align\noexpand\tabu@aligndefault@text
1094             \def\noexpand\tabu@align{\tabu@align}%
1095             \noexpand\fi}%
1096         \aftergroup\noexpand\tabu@usetabu
1097     \noexpand\else
1098         \noexpand\ifdim\tabu@target=\z@
1099             \tabu@target \the\tabu@target\relax
1100         \noexpand\fi
1101         \gdef\noexpand\tabu@preamble{%
1102             \noexpand\ifx \noexpand\tabu@align\noexpand\tabu@aligndefault@text
1103             \def\noexpand\tabu@align{\tabu@align}%
1104             \noexpand\fi}%
1105         \aftergroup\noexpand\tabu@preamble
1106     \noexpand\fi
1107     \the\toks2}%
1108     \tabu@message@save{#1}}%
1109 \fi}%
1110 }% \savetabu
1111 \def\tabu@aligndefault@text{\tabu@aligndefault}%
1112 \def\tabu@warn@usetabu{\PackageWarning{tabu}
1113     {Specifying a target with \string\usetabu space is useless
1114     \MessageBreak The target cannot be changed!}}
1115 \def\tabu@savewd#1{%
1116     \ifnum#1>\tabu@nbcolls \expandafter \@gobble
1117     \else \ifdim\tabu@wd{#1}>\z@
1118         \noexpand\tabu@wddef{#1}{\tabu@wd{#1}}\fi
1119         \expandafter\tabu@savewd % loop
1120     \fi{#1+1}%
1121 }% \tabu@savewd
1122 \def\tabu@make@arstrutbox{%
1123     \@tempdima \ht \strutbox
1124     \advance \@tempdima by\extrarowheight
1125     \setbox \@arstrutbox \hbox{\vrule
1126         \@height \arraystretch \@tempdima
1127         \@depth \arraystretch \dp \strutbox
1128         \@width \z@}%
1129 }% \tabu@make@arstrutbox

```

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

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

```

1130 \def\tabu@savewarn#1#2{\PackageInfo{tabu}
1131     {User-name ‘#1’ already used for \string\savetabu
1132     \MessageBreak #2}}%
1133 \def\tabu@saveerr#1{\PackageError{tabu}
1134     {User-name ‘#1’ is unknown for \string\usetabu
1135     \MessageBreak I cannot restore an unknown preamble!}\@ehd}

```

8.15 \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`.

```

1136 \def\tabu@rowfont{\noalign{\ifnum0='}\fi\tabu@row@font}
1137 \newcommand*\tabu@row@font[2][\{%
1138   \global\let\tabu@rowfontreset \tabu@rowfont@reset
1139   \global\let\tabu@celleft \tabu@celleft
1140   \global\let\tabu@celright \tabu@celright
1141   \gdef\tabu@cellfont{\let\color \tabu@colorleavevmode#2\let\color \tabu@colorORI}%
1142   \ifcsname tabu@cell@#1\endcsname % row alignment
1143     \csname tabu@cell@#1\endcsname \fi
1144   \toks@\expandafter {\tabu@celleft\tabu@cellfont}% inside \noalign group ok
1145   \xdef\tabu@celleft {\the\toks@}%
1146   \ifnum0='}\fi}% end of noalign group
1147 }% \rowfont
1148 \def\tabu@colorleavevmode{\leavevmode\tabu@colorORI}
1149 \AtBeginDocument{\let\tabu@colorORI \color}

```

`\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.

```

1150 \def\tabu@rowfont@reset{%
1151   \noalign{%
1152     \global\let\tabu@rowfontreset \@empty
1153     \global\let\tabu@celleft \tabu@celleft
1154     \global\let\tabu@celright \tabu@celright
1155     \global\let\tabu@cellfont \@empty
1156     \global\let\tabu@celllalign \@empty
1157     \global\let\tabu@cellralign \@empty
1158   }%
1159 }% \tabu@rowfont@reset
1160 \let\tabu@rowfontreset\@empty

```

Preparing stuff to be able to use `\rowfont`

`\tabu@prepnext@tok` `\tabu@prepnext@tok` will replace `\prepnext@tok` (in `array.sty`): its purpose is to add the control sequences `\tabu@celllalign`, `\tabu@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_EX 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: $i = \backslash\text{count@} = \backslash\text{@tempcnta}$: the token to prepare (*ie.* $\backslash\text{toks}<i+1>$) 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@cellalign` 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.

```

1161 \newif\iftabu@cellright
1162 \AtBeginDocument{\let\tabu@prepnext@tokORI \prepnext@tok}% original definition
1163 \def\tabu@prepnext@tok{%
1164   \ifnum \count@<\z@ % <first initialisation>
1165     \@tempcnta \@M % <not initialized by array.sty>
1166     \tabu@nbcolls\z@
1167     \gdef\tabu@global@temp{\tabu@nbcolls\z@}%
1168     \expandafter\let\csname tabu@tok@1L\endcsname \relax
1169     \tabu@cellrightfalse
1170   \else
1171     \ifcase \numexpr \count@-\@tempcnta \relax % (case 0): prev. token is left
1172       \advance \tabu@nbcolls\@ne
1173       \expandafter\let\csname tabu@tok@\the\tabu@nbcolls R\endcsname \relax
1174       \expandafter\gdef\expandafter\tabu@global@temp\expandafter{%
1175         \tabu@global@temp \advance\tabu@nbcolls\@ne}%
1176       \iftabu@cellright % before-previous token is right and is finished
1177         \tabu@cellrightfalse % <only once>
1178         \tabu@savetok R\tabu@preptokenright
1179       \fi
1180       \ifnum \tabu@nbcolls=\@ne \tabu@savetok L\fi
1181       \tabu@preptokenleft
1182     \or % (case 1) previous token is right
1183       \tabu@savetok R\tabu@cellrighttrue
1184     \else % special column: do not change the token
1185       \ifnum \tabu@nbcolls>\z@ %special column: always on the right of normal one
1186         \tabu@savetok R%
1187       \else % unless this is the very first column (\tabu@nbcolls=0)
1188         \advance\tabu@nbcolls\@ne \tabu@savetok L\advance\tabu@nbcolls\m@ne
1189       \fi
1190       \iftabu@cellright % before-previous token is right
1191         \tabu@cellrightfalse
1192         \tabu@preptokenright
1193       \fi
1194     \fi % \ifcase
1195   \fi
1196   \tabu@prepnext@tokORI

```

```

1197 }% \tabu@prepnext@tok
1198 \def\tabu@preptokenright{%
1199     \advance \count@ \m@ne
1200     \toks\count@\expandafter {\the\toks\count@ \tabu@cellright \tabu@celllralign}%
1201     \advance \count@ \@ne
1202 }% \tabu@preptokenright
1203 \def\tabu@preptokenleft{\toks\count@\expandafter {\expandafter\tabu@celllalign
1204                                     \the\toks\count@ \tabu@cellleft}%
1205 }% \tabu@preptokenleft
1206 \def\tabu@savetok#1{\begingroup
1207     \expandafter\tabu@savetok\csname tabu@tok@\the\tabu@nbcols #1\endcsname
1208 }% \tabu@savetok
1209 \def\tabu@savetok#1{%
1210     \ifnum \@chclass=\@ne \toks@\toks\count@
1211     \else\ifnum \@chclass=5 \toks@\toks\count@
1212     \else \toks@{}\fi\fi
1213     \unless\ifx#1\relax
1214         \toks@\expandafter\expandafter\expandafter{%
1215             \expandafter#1\the\toks@}\fi
1216     \toks@\expandafter{\expandafter\def\expandafter#1\expandafter{%
1217                                     \the\toks@}}}%
1218     \@temptokena\expandafter{\tabu@global@temp}%
1219     \xdef\tabu@global@temp {\the\@temptokena \the\toks@}%
1220     \expandafter\endgroup \the\toks@
1221 }% \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> 1222 \let\tabu@cellleft\@empty 1223 \let\tabu@cellright\@empty 1224 \def\tabu@celllalign{\tabu@cellleft}% row font spec. applies to pre-column material 1225 \let\tabu@celllralign\@empty 1226 \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> 1227 \def\tabu@cell@l{% force alignment to left 1228 \gdef\tabu@celllalign{\tabu@removehfil 1229 \raggedright\arraybackslash 1230 \tabu@cellleft}% 1231 \gdef\tabu@celllralign{\tabu@flush1\tabu@ignorehfil}% 1232 \toks@\expandafter{\tabu@cellleft\raggedright\arraybackslash}% local 1233 \xdef\tabu@cellleft{\the\toks@}% 1234 }% \tabu@cell@l 1235 \def\tabu@cell@c{% force alignment to center 1236 \gdef\tabu@celllalign{\tabu@removehfil 1237 \centering\arraybackslash 1238 \tabu@flush{.5}\tabu@cellleft}% 1239 \gdef\tabu@celllralign{\tabu@flush{.5}\tabu@ignorehfil}% 1240 \toks@\expandafter{\tabu@cellleft\centering\arraybackslash}% local 1241 \xdef\tabu@cellleft{\the\toks@}% 1242 }% \tabu@cell@c 1243 \def\tabu@cell@r{% force alignment to right 1244 \gdef\tabu@celllalign{\tabu@removehfil 1245 \raggedleft\arraybackslash </pre>
--	---

```

1246             \tabu@flush1\tabu@cellleft}%
1247 \gdef\tabu@celllralign{\tabu@ignorehfil}%
1248 \toks@\expandafter{\tabu@cellleft\raggedleft\arraybackslash}% local
1249 \xdef\tabu@cellleft{\the\toks@}%
1250 }% \tabu@cell@r
1251 \def\tabu@cell@j{% force justification (for p, m, b columns)
1252 \gdef\tabu@celllalign{\tabu@justify\tabu@cellleft}%
1253 \global\let\tabu@celllralign\@empty
1254 \toks@\expandafter{\tabu@cellleft\tabu@justify}% local (noalign grp)
1255 \xdef\tabu@cellleft{\the\toks@}%
1256 }% \tabu@cell@j
1257 \def\tabu@justify{%
1258 \leftskip\z@skip \@rightskip\leftskip \rightskip\@rightskip
1259 \parfillskip\@flushglue
1260 }% \tabu@justify
1261 %% ragged2e settings
1262 \def\tabu@cell@L{% force alignment to left (ragged2e)
1263 \gdef\tabu@celllalign{\tabu@removehfil
1264 \RaggedRight\arraybackslash
1265 \tabu@cellleft}%
1266 \gdef\tabu@celllralign{\tabu@flush1\tabu@ignorehfil}%
1267 \toks@\expandafter{\tabu@cellleft\RaggedRight\arraybackslash}%
1268 \xdef\tabu@cellleft{\the\toks@}%
1269 }% \tabu@cell@L
1270 \def\tabu@cell@C{% force alignment to center (ragged2e)
1271 \gdef\tabu@celllalign{\tabu@removehfil
1272 \Centering\arraybackslash
1273 \tabu@flush{.5}\tabu@cellleft}%
1274 \gdef\tabu@celllralign{\tabu@flush{.5}\tabu@ignorehfil}%
1275 \toks@\expandafter{\tabu@cellleft\Centering\arraybackslash}%
1276 \xdef\tabu@cellleft{\the\toks@}%
1277 }% \tabu@cell@C
1278 \def\tabu@cell@R{% force alignment to right (ragged2e)
1279 \gdef\tabu@celllalign{\tabu@removehfil
1280 \RaggedLeft\arraybackslash
1281 \tabu@flush1\tabu@cellleft}%
1282 \gdef\tabu@celllralign{\tabu@ignorehfil}%
1283 \toks@\expandafter{\tabu@cellleft\RaggedLeft\arraybackslash}%
1284 \xdef\tabu@cellleft{\the\toks@}%
1285 }% \tabu@cell@R
1286 \def\tabu@cell@J{% force justification (ragged2e)
1287 \gdef\tabu@celllalign{\justifying\arraybackslash\tabu@cellleft}%
1288 \global\let\tabu@celllralign\@empty
1289 \toks@\expandafter{\tabu@cellleft\justifying\arraybackslash}%
1290 \xdef\tabu@cellleft{\the\toks@}%
1291 }% \tabu@cell@J
1292 \def\tabu@flush#1{%
1293 \iftabu@colortbl % colortbl uses \hfill rather than \hfil
1294 \hskip \ifnum\currentgrouptype>13 \stretch{#1}%
1295 \else\ifdim#1pt<1pt \tabu@cellskip
1296 \else \stretch{#1}
1297 \fi\fi \relax
1298 \else % array.sty
1299 \ifnum \currentgrouptype>13\relax
1300 \hfil \hskip1sp
1301 \fi
1302 \fi
1303 }% \tabu@flush

```

```

1304 \AtBeginDocument{%
1305   \@ifpackageloaded{ragged2e}
1306   {}
1307   {\let\tabu@cell@L \tabu@cell@l
1308    \let\tabu@cell@R \tabu@cell@r
1309    \let\tabu@cell@C \tabu@cell@c
1310    \let\tabu@cell@J \tabu@cell@j
1311   }%
1312 }% 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.

```

1313 \newskip\tabu@cellskip
1314 \let\tabu@hfil\hfil
1315 \let\tabu@hfill\hfill
1316 \let\tabu@hskip\hskip
1317 \def\tabu@removehfil{%
1318   \iftabu@colortbl
1319     \unkern \tabu@cellskip = \lastskip
1320     \ifnum\gluestretchorder\tabu@cellskip = \tw@ \hskip-\tabu@cellskip
1321     \else \tabu@cellskip = \z@skip
1322     \fi
1323   \else
1324     \ifdim\lastskip=1sp\unskip\fi
1325     \ifnum\gluestretchorder\lastskip = \@ne
1326       \hfilneg % \hfilneg for array.sty but not for colortbl...
1327     \fi
1328   \fi
1329 }% \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.

```

1330 \def\tabu@ignorehfil{%
1331   \aftergroup\tabu@nohfil
1332 }% \tabu@ignorehfil
1333 \def\tabu@nohfil{% \hfil -> do nothing + restore original \hfil
1334   \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1335 }% \tabu@nohfil
1336 \AtBeginDocument{%
1337   \@ifpackageloaded{colortbl}
1338   {%
1339     \def\tabu@nohfil{%
1340       \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1341       \def\hfill{\let\hfill\tabu@hfill}% (colortbl uses \hfill) pfff...
1342       \def\hskip##1\relax{\let\hskip\tabu@hskip}}% local
1343     }% \@ifpackageloaded colortbl
1344   }%
1345 }% AtBeginDocument

```

8.16 Utilities

`tabu \fbox`

`\tabu@fbox` works exactly like L^AT_EX `\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.

```

1346 \def\tabu@fbox{%
1347   \leavevmode

```

```

1348 \let\color@bgroup\bgroup
1349 \def\color@egroup{\endgraf\egroup}%
1350 \afterassignment\tabu@begin@fbox
1351 \setbox\@tempboxa \hbox
1352 }% \tabu@fbox
1353 \def\tabu@begin@fbox{\color@bgroup\kern\fboxsep\aftergroup\tabu@end@fbox}
1354 \def\tabu@end@fbox{\kern\fboxsep\color@egroup\@frameb@x\relax}

```

\centering, \raggedright, \raggedleft

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

```

1355 \expandafter\def\expandafter\tabu@centering\expandafter{%
1356                                     \centering\arraybackslash}
1357 \expandafter\def\expandafter\tabu@raggedleft\expandafter{%
1358                                     \raggedleft\arraybackslash}
1359 \expandafter\def\expandafter\tabu@raggedright\expandafter{%
1360                                     \raggedright\arraybackslash}
1361 \def\tabu@trivlist{\expandafter\def\expandafter\@trivlist\expandafter{%
1362 \expandafter\let\expandafter\\ \expandafter\@centercr\@trivlist}}%

```

8.17 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.

```

1363 \def\tabu@fix@arrayright{%
1364 \ifpackageloaded{arydshln}
1365 {%
1366 \ifpackageloaded{colortbl}
1367 {%% colortbl + arydshln
1368 \def\tabu@endarray{%
1369 \adl@endarray \egroup \adl@arrayrestore \CT@end \egroup %<original>
1370 \@arrayright % <FC>
1371 \gdef\@preamble{}}% <FC>
1372 }}% \endarray
1373 {%% arydshln / no colortbl
1374 \def\tabu@endarray{%
1375 \adl@endarray \egroup \adl@arrayrestore \egroup %<original>
1376 \@arrayright % <FC>
1377 \gdef\@preamble{}}% <FC>
1378 }}% \endarray
1379 }%
1380 {%
1381 \ifpackageloaded{colortbl}
1382 {%% colortbl / no arydshln
1383 \def\tabu@endarray{%
1384 \crrc \egroup \egroup
1385 \@arrayright % <FC>
1386 \gdef\@preamble{\CT@end
1387 }}%
1388 {\PackageWarning{tabu}
1389 {\string\@arrayright\space is missing from the
1390 \MessageBreak definition of \string\endarray.

```

```

1391      \MessageBreak Comptability with delarray.sty is broken.}}%
1392    }%
1393 }% \tabu@fix@arrayright

```

arydshln @ columns

```

1394 \def\tabu@adl@xarraydashrule#1#2#3{%
1395     \ifnum\@lastchclass=\adl@class@start\else
1396     \ifnum\@lastchclass=\@ne\else
1397     \ifnum\@lastchclass=5 \else % <FC> @-arg (class 5) and !-arg (class 1)
1398         \adl@leftrulefalse \fi\fi          % must be treated the same
1399     \fi
1400     \ifadl@zwvrule\else \ifadl@inactive\else
1401         \addtopreamble{\vrule\@width\arrayrulewidth
1402             \@height\z@ \@depth\z@}\fi \fi
1403     \ifadl@leftrule
1404         \addtopreamble{\adl@vlineL{\CT@arc@}{\adl@dashgapcolor}%
1405             {\number#1}#3}%
1406     \else \addtopreamble{\adl@vlineR{\CT@arc@}{\adl@dashgapcolor}%
1407         {\number#2}#3}
1408     \fi
1409 }% \tabu@adl@xarraydashrule

```

arydshln, colors without colortbl and empty p columns

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

```

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

```

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

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

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

Fixed inside tabu.


```

1410 \def\tabu@adl@act@endpbox{%
1411     \unskip \ifhmode \nobreak \fi \@finalstrut \@arstrutbox
1412     \egroup \egroup
1413     \adl@colhtdp \box\adl@box \hfil
1414 }% \tabu@adl@act@endpbox
1415 \def\tabu@adl@fix{%
1416     \let\adl@xarraydashrule \tabu@adl@xarraydashrule % <fix> arydshln
1417     \let\adl@act@endpbox \tabu@adl@act@endpbox % <fix> arydshln
1418     \let\adl@act@@endpbox \tabu@adl@act@endpbox % <fix> arydshln
1419     \let\@preamerror \@preamerr % <fix> arydshln
1420 }% \tabu@adl@fix

```

1421 </package>

9 References

- [1] *A new implementation of L^AT_EX's tabular and array environments* by Frank Mittelbach
2008/09/09 v2.4c – Tabular extension package (FMi)
CTAN:help/Catalogue/entries/array.html
- [2] *The varwidth package* by Donald Arseneau
2009/03/30 ver 0.92 – Variable-width minipages
CTAN:help/Catalogue/entries/varwidth.html
- [3] *The enumitem-zref package* by 
2010/11/28 ver 1.1 – Extended references for enumitem pkg
CTAN:help/Catalogue/entries/enumitem-zref.html

10 History

[2010/12/18 v1.7]

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

[2010/12/07 v1.5]

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

[2010/11/22 v1.4]

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

[2010/11/18 v1.3]

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

[2010/11/15 v1.2]

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

[2010/10/28 v1.1]

- First version.

11 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>\@arrayright</code>	55, 480, 595, 1370, 1376, 1385, 1389
<code>\@arstrutbox</code>	80, 737, 780, 786, 792, 846, 847, 871, 1125, 1411
<code>\@centercr</code>	211, 1362
<code>\@chclass</code>	1210, 1211
<code>\@currentHref</code>	315
<code>\@currentlabelname</code>	316
<code>\@defaultunits</code>	880, 881, 888, 907, 910
<code>\@elt</code>	484, 485
<code>\@finalstrut</code>	737, 1411
<code>\@firstoftwo</code>	248, 716
<code>\@flushglue</code>	151, 154, 155, 725, 1259
<code>\@footnotemark</code>	288
<code>\@footnotetext</code>	208, 479
<code>\@gobbletwo</code>	694
<code>\@halignto</code>	89, 129, 141, 242, 243, 252, 253, 472, 597, 1088
<code>\@ifnextchar</code>	26, 83
<code>\@let@token</code>	227, 229, 230, 231, 232, 233, 374, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392
<code>\@m</code>	674
<code>\@makeother</code>	1001
<code>\@mpfn</code>	276, 285, 294
<code>\@multicnt</code>	842, 848, 852, 853, 855, 860, 861, 863, 864, 868
<code>\@preamble</code>	455, 496, 502, 506, 1065, 1090, 1371, 1377, 1386
<code>\@preamerr</code>	1419
<code>\@preamerror</code>	1419
<code>\@rightskip</code>	1258
<code>\@sanitize</code>	994
<code>\@secondoftwo</code>	719
<code>\@spaces</code>	669
<code>\@sptoken</code>	146, 231, 389, 957, 1012, 1046
<code>\@tempcnta</code>	474, 1165, 1171
<code>\@tempdimb</code>	913, 916, 921, 930, 931, 933, 935, 942, 946
<code>\@tempdimc</code>	926, 931, 932, 933, 936, 938, 939, 948, 949
<code>\@temptokena</code>	767, 1004, 1020, 1024, 1055, 1058, 1218, 1219
<code>\@thefnmark</code>	286, 296
<code>\@trivlist</code>	1361, 1362
<code>\@undefined</code>	7, 461
<code>\@whilenum</code>	788, 848, 868
<code>\@whilesw</code>	603
<code>\@xfootnote</code>	209, 281, 479
<code>\@xfootnotetext</code>	208
<code>\{</code>	1001
<code>\}</code>	1001
<code>\^</code>	527, 993, 994, 995
<code>_</code>	995
A	
<code>\adl@act@@endpbox</code>	1418
<code>\adl@act@endpbox</code>	1417

<code>\adl@arrayrestore</code>	1369, 1375	<code>\extrarowheight</code>	267, 688, 786, 899, 944, 945, 988, 989, 1124
<code>\adl@box</code>	1413	<code>\extratabsurround</code>	268, 689
<code>\adl@class@start</code>	1395	F	
<code>\adl@colhtdp</code>	1413	<code>\footnotetext</code>	276, 280
<code>\adl@dashgapcolor</code>	1404, 1406	<code>\futurelet</code>	144, 227, 245, 374, 815, 953, 960, 961, 1008
<code>\adl@endarray</code>	1369, 1375	G	
<code>\adl@leftrulefalse</code>	1398	<code>\GenericError</code>	171, 478
<code>\adl@ncol</code>	553	<code>\gluestretchorder</code>	1320, 1325
<code>\adl@vlineL</code>	1404	H	
<code>\adl@vlineR</code>	1406	<code>\HC@footnotetext</code>	308
<code>\adl@xarraydashrule</code>	1416	<code>\hbadness</code>	474
<code>\afterassignment</code> 27, 165, 244, 422, 767, 906, 1350		<code>\hfilneg</code>	1326
<code>\aftergroup</code>	112, 113, 132, 133, 134, 137, 161, 162, 169, 173, 198, 225, 726, 1096, 1105, 1331, 1353	<code>\hfuzz</code>	197, 475
<code>\array</code>	176	<code>\ht</code>	786, 847, 1123
<code>\arraybackslash</code>	724, 1229, 1232, 1237, 1240, 1245, 1248, 1264, 1267, 1272, 1275, 1280, 1283, 1287, 1289, 1356, 1358, 1360	<code>\Hy@footnote@currentHref</code>	299
<code>\arraycolsep</code>	263	<code>\Hy@raisedlink</code>	312
<code>\arrayrulewidth</code>	265, 686, 813, 821, 912, 915, 919, 986, 988, 1401	<code>\hyper@anchor</code>	310, 313
<code>\arraystretch</code>	269, 691, 1126, 1127	I	
<code>\AtEndOfPackage</code>	7, 30, 33, 38, 41	<code>\ifadl@inactive</code>	1400
B		<code>\ifadl@leftrule</code>	1403
<code>\begin</code>	4, 4	<code>\ifadl@zwvrule</code>	1400
C		<code>\ifcase</code>	229, 377, 459, 560, 770, 855, 860, 911, 955, 1011, 1028, 1079, 1171, 1194
<code>\c@footnote</code>	282	<code>\ifcat</code>	121, 392, 820, 911, 958
<code>\c@page</code>	344	<code>\ifcsname</code>	322, 603, 710, 893, 968, 1142
<code>\catcode</code>	12, 14, 527, 993, 994, 995, 997, 998	<code>\ifdim</code>	138, 199, 203, 252, 339, 430, 433, 451, 452, 560, 561, 563, 564, 573, 577, 581, 585, 596, 604, 607, 610, 621, 624, 628, 632, 647, 717, 739, 743, 745, 746, 930, 931, 932, 933, 934, 935, 936, 938, 942, 946, 948, 1080, 1098, 1117, 1295, 1324
<code>\Centering</code>	402, 1272, 1275	<code>\iffalse</code>	456
<code>\centering</code>	209, 399, 1237, 1240, 1356	<code>\ifHy@nesting</code>	309
<code>\chardef</code>	80	<code>\ifin@</code>	56, 356, 708, 716, 718
<code>\cl@@ckpt</code>	485	<code>\ifnum</code>	28, 456, 491, 501, 565, 584, 594, 662, 672, 673, 674, 676, 677, 786, 838, 843, 852, 855, 860, 863, 874, 882, 883, 884, 885, 1116, 1136, 1146, 1164, 1180, 1185, 1210, 1211, 1294, 1299, 1320, 1325, 1395, 1396, 1397
<code>\color</code>	969, 1141, 1149	<code>\iftabu@cellright</code>	1161, 1176, 1190
<code>\color@</code>	968	<code>\iftabu@colortbl</code>	164, 270, 781, 794, 812, 814, 832, 898, 927, 983, 1068, 1293, 1318
<code>\color@bgroup</code>	1348, 1353	<code>\iftabu@firstcline</code>	75, 847, 940, 985
<code>\color@egroup</code>	1349, 1354	<code>\iftabu@measuring</code>	75, 330, 520, 546, 590
<code>\copy</code>	892	<code>\iftabu@negcoef</code>	75, 132, 459, 559, 702, 1075
<code>\crrc</code>	553, 779, 1384	<code>\iftabu@spread</code>	75, 337, 459, 517, 543, 715
<code>\CT@drsc@</code>	1072	<code>\ignorespaces</code>	1011, 1045
<code>\currentgrouptype</code>	1294, 1299	<code>\immediate</code>	167, 169, 173
D		<code>\in@</code>	55, 356, 707
<code>\d@llarend</code>	121	<code>\in@false</code>	717
<code>\DEBUG</code>	6	J	
<code>\delarray_(package_option)</code>	37	<code>\justifying</code>	1287, 1289
<code>\detokenize</code>	692, 893, 894, 976	L	
<code>\dimen@</code>	558, 563, 581, 599	<code>\lastbox</code>	202
<code>\doublerulesep</code>	266, 687	<code>\lastskip</code>	1319, 1324, 1325
<code>\dp</code>	1127	<code>\leaders</code>	984
E		<code>\leftskip</code>	1258
<code>\endarray</code> 55, 56, 176, 190, 542, 1372, 1378, 1390			
<code>\endlinechar</code>	149, 152, 799, 996		
<code>\endlongtable</code>	182		
<code>\endlongtabu</code>	182		
<code>\endtabu</code>	175		
<code>\endtabular</code>	177		
<code>\everycr</code>	215, 766, 767		
<code>\everyeof</code>	1002		
<code>\everypar</code>	200, 201		
<code>\extracolsep</code>	139, 151, 154, 155		

\light_{(package_option)}	40	\scantokens	149, 152, 799, 1002
\linebreak	995	\set@color	969
\linegoal	35, 343	\setbox ...	202, 493, 505, 514, 722, 1125, 1351
\linegoal_{(package_option)}	31	\setlength	340
\linewidth	31, 35, 195, 199, 203, 342, 477	\span	866
\LNGL@setlinegoal	247, 248	\stretch	1294, 1296
\longtable	182	\strip@pt	333, 348, 437
\longtabu	61, 180, 186	\strutbox	1123, 1127
\loop	850		
\lower	899		
\LT@bchunk	499, 503, 508		
\LT@echunk	513		
\LT@get@widths	515		
\LT@output	509		
\LTchunksizes	507		
	M		T
\maxdimen	475	\tabcolsep	264
\message	6, 24, 29, 117, 120	\tabu	175
\minrowclearance	270, 690, 782, 783, 795	\tabu@	176, 177, 182, 259
	N	\tabu@@begin	254, 256, 260
\narrowragged	724	\tabu@@cellleft	1139, 1153
\NC@find 90, 123, 128, 151, 154, 155, 271, 692, 807		\tabu@@cellright	1140, 1154
\NC@rewrite@X	111	\tabu@@rewrite@X	355, 723
\newcolumnntype	54	\tabu@@tabudecimal	1026
\newcount	63, 64, 65, 66, 67, 68	\tabu@add	853, 861, 876
\newdimen	69, 70, 71, 72, 73, 74	\tabu@adl@act@endpbox	1410, 1414, 1417, 1418
\newsavebox	79	\tabu@adl@endTRIAL	60, 541, 553
\newskip	1313	\tabu@adl@fix	60, 190, 1415, 1420
\newtoks	273	\tabu@adl@xarraydashrule .	1394, 1409, 1416
\noalign	553, 769, 779,	\tabu@align 258, 259, 847, 1093, 1094, 1102, 1103	
791, 838, 846, 871, 1061, 1136, 1144, 1151		\tabu@aligndefault	195, 197, 254, 1111
\null	995	\tabu@aligndefault@text ...	1093, 1102, 1111
\numexpr	1076, 1171	\tabu@arith	518, 544, 554
	O	\tabu@arith@negcoef	559, 601, 616
\omit	851	\tabu@arrayleft@measure	183, 469, 489
\on@line	344, 661, 696	\tabu@arstrutbox	80, 792, 871
	P	\tabu@begin	227
\PackageError	48, 50, 181, 187, 1133	\tabu@begin@fbox	1350, 1353
\PackageInfo	1130	\tabu@box 79, 493, 505, 516, 560, 561, 570,	
\PackageWarning 174, 412, 417, 900, 1112, 1388		575, 582, 584, 621, 623, 632, 643, 648,	
\parfillskip	725, 995, 1259	652, 657, 722, 737, 738, 741, 746, 747, 754	
\pdfadjustspacing	731, 732	\tabu@cell@C	1270, 1277, 1309
\preamble	13, 103, 109, 114	\tabu@cell@c	1227
\preamble_{(new_column_type)}	103	\tabu@cell@J	1286, 1291, 1310
\prepnext@tok	42, 214, 1162	\tabu@cell@j	1227
\ProcessOptions	52	\tabu@cell@L	1262, 1269, 1307
\protected	168, 172	\tabu@cell@l	1227
\protected@xdef	286, 295	\tabu@cell@R	1278, 1285, 1308
	R	\tabu@cell@r	1227
\RaggedLeft	403, 1280, 1283	\tabu@cellfont	1141, 1144, 1155, 1222
\raggedleft	210, 400, 1245, 1248, 1358	\tabu@cellllalign	1156, 1203, 1222,
\RaggedRight	401, 1264, 1267	1228, 1236, 1244, 1252, 1263, 1271, 1279	
\raggedright	210, 398, 1229, 1232, 1360	\tabu@cellllalign	1287
\remove@to@nnil	394, 963, 970	\tabu@cellleft	1044, 1139, 1144,
\repeat	867	1145, 1153, 1204, 1222, 1230, 1232, 1233,	
\rightskip	1258	1238, 1240, 1241, 1246, 1248, 1249, 1252,	
\rowfont	13, 49, 213, 214, 1136	1254, 1255, 1265, 1267, 1268, 1273, 1275,	
	S	1276, 1281, 1283, 1284, 1287, 1289, 1290	
\save@decl	42, 122	\tabu@cellralign ...	1157, 1200, 1222, 1231,
\savetabu 12, 257, 477, 593, 696, 697, 1061, 1131		1239, 1247, 1253, 1266, 1274, 1282, 1288	
		\tabu@cellright	1140, 1154, 1200, 1222
		\tabu@cellrightfalse	1169, 1177, 1191
		\tabu@cellrighttrue	1183
		\tabu@cellskip ..	1295, 1313, 1319, 1320, 1321
		\tabu@centering	209, 1355
		\tabu@ccline	839, 840, 841, 875
		\tabu@cnt 63, 422, 425, 468, 555, 565, 584, 676	
		\tabu@collect	497, 510, 527
		\tabu@collectbody	527, 528, 535, 537
		\tabu@color	819,
			831, 833, 906, 909, 920, 927, 928, 937, 969

<code>\tabu@colorleavevmode</code>	1141, 1148	<code>\tabu@linegoal@target</code>	245, 246, 250
<code>\tabu@colorORI</code>	1141, 1148, 1149	<code>\tabu@linegoaltarget</code>	244, 245
<code>\tabu@colortblfalse</code>	58	<code>\tabu@lineoff</code>	914, 917, 922, 925, 952
<code>\tabu@colortbltrue</code>	58	<code>\tabu@lineon</code>	905, 908, 924
<code>\tabu@colortest</code>	906, <u>953</u>	<code>\tabu@lines</code>	44, 115, <u>799</u>
<code>\tabu@dashgap</code>	837, 914, 917, 931	<code>\tabu@lines@</code>	800, 802, 805
<code>\tabu@debug</code>	22, 29, 748, 872	<code>\tabu@linestyle</code>	973, 974, 980
<code>\tabu@decimal</code>	1004, 1058	<code>\tabu@long@arrayleft@measure</code>	183, <u>499</u>
<code>\tabu@defaultleaders</code>	889, 903, 982, 983	<code>\tabu@long@endTRIAL</code>	184, 512, 526
<code>\tabu@defleaders</code>	890, 891, 892, 897	<code>\tabu@make@arstrutbox</code>	1091, 1122, 1129
<code>\tabu@dimexpr</code>	112, 365, 447, 450, 453, 707, 716, 719	<code>\tabu@makeatletter</code>	994
<code>\tabu@end@fbox</code>	1353, 1354	<code>\tabu@maybecolor</code>	827, 906, 912, 913, 916, 919, 926
<code>\tabu@endarray</code>	56, 190, 1368, 1374, 1383	<code>\tabu@measure@endpbox</code>	<u>705</u>
<code>\tabu@endenvir</code>	524, 531, 550	<code>\tabu@measure@startpbox</code>	466, 556, 646, <u>705</u>
<code>\tabu@endTRIAL</code>	184, 494, 508, <u>540</u>	<code>\tabu@measuringfalse</code>	216, 520, 546
<code>\tabu@everycr</code>	215, 766, 768	<code>\tabu@measuringtrue</code>	336, 566, 573, 580, 639
<code>\tabu@fbox</code>	213, 1346, 1352	<code>\tabu@message</code>	20, 24, 341, 345, 555, 563, 564, 565, 569, 574, 609, 620, 627, 628, 631, 632, 695
<code>\tabu@first@cline</code>	765, 770, 771, 772, 774, 796	<code>\tabu@message@arith</code>	555, 574, <u>641</u>
<code>\tabu@firstcline</code>	44	<code>\tabu@message@negcoef</code>	609, <u>641</u>
<code>\tabu@firstclinefalse</code>	773	<code>\tabu@message@reached</code>	563, 564, 565, 655
<code>\tabu@firstclinettrue</code>	764	<code>\tabu@message@save</code>	21, 25, 1108
<code>\tabu@fix@arrayright</code>	57, 1363, 1393	<code>\tabu@message@spread</code>	620, <u>641</u>
<code>\tabu@flush</code>	1231, 1238, 1239, 1246, 1266, 1273, 1274, 1281, 1292, 1303	<code>\tabu@minrowclearance</code>	782, 795
<code>\tabu@footnotetext</code>	<u>274</u>	<code>\tabu@msg@align</code>	631, 642, 643, 644, 645, 648, 651, 652, 653, 654, 656, 657, 658, 671
<code>\tabu@footnotes</code>	198, <u>273</u> , 275, 277, 279, 292, 302	<code>\tabu@naturalX</code>	<u>69</u> , 222, 472, 618, 623, 654, 712, 738, 739, 740
<code>\tabu@footnotetext</code>	208, 274, 324	<code>\tabu@naturalX@max</code>	224, 463, 618, 739, 740, 762
<code>\tabu@GenericError</code>	168, 478	<code>\tabu@naturalX@min</code>	223, 464, 619, 743, 744, 761
<code>\tabu@GenericErrorORI</code>	170, 171	<code>\tabu@nbcols</code>	<u>63</u> , 553, 788, 868, 878, 882, 884, 885, 1116, 1166, 1167, 1172, 1173, 1175, 1180, 1185, 1187, 1188, 1207
<code>\tabu@get@decimal</code>	1020, 1048	<code>\tabu@NC@list</code>	119, 125, 127, 217
<code>\tabu@get@decimalspace</code>	1023, 1050	<code>\tabu@NC@rewrite@X</code>	<u>88</u> , 111
<code>\tabu@getdecimal</code>	1014, <u>1027</u>	<code>\tabu@negcoeffalse</code>	216
<code>\tabu@getdecimal@</code>	1005, 1008, 1014	<code>\tabu@negcoeftrue</code>	132, 430
<code>\tabu@getdecimal@ignorespaces</code>	1005, 1010, 1019	<code>\tabu@nested</code>	68, 160, 162, 194, 220, 225, 594, 662, 677, 756, 758, 760, 786
<code>\tabu@getline</code>	895, <u>905</u>	<code>\tabu@newcolumnntype</code>	<u>91</u> , 95, 103, 110, 126, 803
<code>\tabu@getlinecolor</code>	822, 962, 966, 971	<code>\tabu@next</code>	128, 139, 142, 151, 154, 155, 156, 960, 961, 962, 964
<code>\tabu@getlinespec</code>	842, <u>887</u> , 975	<code>\tabu@noadjustspacing</code>	725, 732, 734
<code>\tabu@gettarget</code>	242, 243, 244	<code>\tabu@nocline</code>	43, 50
<code>\tabu@global@temp</code>	112, 191, 1167, 1174, 1175, 1218, 1219	<code>\tabu@nohfil</code>	1331, 1333, 1335, 1339
<code>\tabu@global@X</code>	160, 161, 191, 331, 346, 471, 485, 492, 503, 507, 591	<code>\tabu@nolongtabu</code>	61, 187
<code>\tabu@global@Xnested</code>	131, 159, 163, 491, 501	<code>\tabu@norowfont</code>	43, 48
<code>\tabu@gobble@pt</code>	83, 84	<code>\tabu@nowrite</code>	<u>165</u> , 478
<code>\tabu@gobbleopt</code>	<u>82</u> , 359	<code>\tabu@preamble</code>	1101, 1105
<code>\tabu@gobblespace</code>	<u>82</u> , 147, 236, 961, 1016	<code>\tabu@prep@TRIAL</code>	133, <u>454</u>
<code>\tabu@halignto</code>	130, 141, 253, 271, 352, 637	<code>\tabu@preparesave</code>	257, <u>261</u>
<code>\tabu@header</code>	569, 641, 655, 676	<code>\tabu@prepnext@tok</code>	42, 214, <u>1161</u>
<code>\tabu@hfil</code>	1314, 1334, 1340	<code>\tabu@prepnext@tokORI</code>	1162, 1196
<code>\tabu@hfill</code>	1315, 1341	<code>\tabu@preptokenleft</code>	1181, 1203, 1205
<code>\tabu@hfuzz</code>	197, 560, 571, 659, 661	<code>\tabu@preptokenright</code>	1178, 1192, 1198, 1202
<code>\tabu@hskip</code>	1316, 1342	<code>\tabu@printdecimal</code>	1051, 1054, 1060
<code>\tabu@Hy@footnotetext</code>	297, 307, 320	<code>\tabu@pushbegins</code>	529, 538, 539
<code>\tabu@Hy@fnttext</code>	<u>282</u>	<code>\tabu@raggedleft</code>	210, 1357
<code>\tabu@Hy@fnttxt</code>	282, 288, 290, 306	<code>\tabu@raggedright</code>	210, 1359
<code>\tabu@Hy@xfootnote</code>	<u>282</u>	<code>\tabu@removehfil</code>	1228, 1236, 1244, 1263, 1271, 1279, <u>1313</u>
<code>\tabu@ignorehfil</code>	1231, 1239, 1247, 1266, 1274, 1282, <u>1330</u>	<code>\tabu@restore@NC@list</code>	113, 125
<code>\tabu@immediate</code>	167, 170, 174	<code>\tabu@restoreeverycr</code>	766, 767
<code>\tabu@justify</code>	1252, 1254, 1257, 1260	<code>\tabu@rewrite@last</code>	144, 145
<code>\tabu@lasttry</code>	468, 561, 582, 584		
<code>\tabu@leaderfill</code>	899, 950, 991, 992		
<code>\tabu@linecolor</code>	953, 954, 960, 961, 965		
<code>\tabu@linedash</code>	836, 905, 933		

<code>\tabu@rewrite@X</code>	88, 99, 328 , 476, 723	<code>\tabu@TRIAL</code>	489 , 504, 521, 532, 547
<code>\tabu@rewrite@Xnested</code>	356 , 476	<code>\tabu@TRIAL@hook</code>	481, 487, 488
<code>\tabu@rewrite@Xrestore</code>	99, 356	<code>\tabu@trivlist</code>	211, 1361
<code>\tabu@rewritefirst</code>	110 , 124, 217, 259	<code>\tabu@usetabu</code>	1086, 1096
<code>\tabu@rewritefirst_(new_column_type)</code>	110	<code>\tabu@vcenter</code>	81, 793
<code>\tabu@rewritelast</code>	116,	<code>\tabu@verb</code>	994, 1000
	119 , 123 , 126 , 139 , 143 , 144 , 147 , 158	<code>\tabu@vline</code>	808, 813, 834
<code>\tabu@rewritelast_(new_column_type)</code>	126	<code>\tabu@warn</code>	172, 570, 578
<code>\tabu@row@font</code>	1136, 1137	<code>\tabu@warn@usetabu</code>	1081, 1112
<code>\tabu@rowfont</code>	43, 213, 1136	<code>\tabu@wd</code>	451, 452, 604, 607, 610, 614,
<code>\tabu@rowfont@reset</code>	1138, 1150		665, 717, 745, 746, 751, 752, 756, 1117, 1118
<code>\tabu@rowfontreset</code>	776, 1138, 1152, 1160	<code>\tabu@wddef</code>	693, 747, 757, 1118
<code>\tabu@sanitizetext</code>	192, 212, 993	<code>\tabu@wdlet</code>	362, 432, 759
<code>\tabu@save@decl</code>	42, 85 , 122	<code>\tabu@write</code>	166, 170, 174
<code>\tabu@savecounters</code>	473, 483, 486	<code>\tabu@X@cols</code> 63 , 111, 221, 329, 332, 333, 347,	
<code>\tabu@saveditparam</code>	262, 700, 1066		361, 362, 365, 432, 447, 460, 461, 462, 602,
<code>\tabu@saveditpream</code>	271, 701, 1067		603, 604, 606, 607, 609, 610, 612, 709,
<code>\tabu@saveditpreamble</code>	455, 496, 502, 506		710, 711, 742, 745, 746, 747, 749, 751, 752
<code>\tabu@saveerr</code>	96, 98, 104, 106, 1130	<code>\tabu@X@sum</code>	72 , 333,
<code>\tabu@savetok</code>	1207, 1209		334, 348, 349, 364, 368, 425, 430, 431,
<code>\tabu@savetok</code>	1178,		433, 437, 558, 563, 564, 567, 573, 575,
	1180 , 1183 , 1186 , 1188 , 1206, 1208, 1221		576, 581, 596, 599, 608, 645, 647, 649, 741
<code>\tabu@savewarn</code>	1062, 1063, 1130	<code>\tabu@Xalign</code> 398, 399, 400, 401, 402, 403, 411, 415	
<code>\tabu@savewd</code> 702, 1076, 1077, 1115, 1119, 1121		<code>\tabu@Xarg</code>	328, 360
<code>\tabu@scandecimal</code>		<code>\tabu@Xc@ef</code>	422, 424, 427
	1006, 1008, 1009, 1016, 1021, 1024	<code>\tabu@Xcoef</code>	405, 407, 421, 423
<code>\tabu@settarget</code>	178, 185, 227	<code>\tabu@Xfinish</code>	490, 500, 523, 549
<code>\tabu@setup</code>	178, 185, 189	<code>\tabu@xfootnote</code>	209, 281, 325
<code>\tabu@setup@everycr</code>		<code>\tabu@xfootnotetext</code>	208, 278
	45, 134, 137, 495, 502, 506, 763	<code>\tabu@xleaders</code>	844,
<code>\tabu@setup@saveglobal</code>	193, 219, 226		858, 887, 889, 898, 903, 937, 977, 978, 982
<code>\tabu@setup@TRIAL</code>	457, 471	<code>\tabu@Xmath</code>	370, 406, 439, 443
<code>\tabu@skipdecimal</code>	1009, 1015, 1017, 1049	<code>\tabu@Xparse</code>	360
<code>\tabu@spaces</code>	646, 660, 663, 669, 748	<code>\tabu@Xparsespace</code>	375, 404
<code>\tabu@spread</code>	238, 243	<code>\tabu@Xtest</code>	374, 376, 410
<code>\tabu@spreadarith</code>	517, 543, 617	<code>\tabu@Xtype</code>	395, 396, 397, 416, 420
<code>\tabu@spreadfalse</code>	216, 639	<code>\tabu@cline</code>	8, 43, 51, 214, 836
<code>\tabu@spreadheader</code>	650, 680	<code>\tabu@cline*</code>	9
<code>\tabu@spreadtarget</code>	69 ,	<code>\tabu@cline@warn</code>	900, 904, 920
	337, 338, 351, 621, 622, 624, 625, 628, 651	<code>\tabu@colX</code> 72 , 357, 358, 365, 366, 437, 458, 567,	
<code>\tabu@spreadtrue</code>	243		573, 575, 577, 579, 585, 592, 607, 619,
<code>\tabu@start</code>	66, 843, 848, 880, 882, 883		623, 638, 642, 653, 656, 667, 698, 1084, 1089
<code>\tabu@start@stop</code>	878, 879, 886	<code>\tabu@colX@align</code>	369, 412, 414, 436
<code>\tabu@startpboxORI</code>	465, 557, 636, 706	<code>\tabu@colX@do</code>	360
<code>\tabu@startstop</code>	842, 878	<code>\tabu@colX@error</code>	72 , 579
<code>\tabu@stop</code> 67, 843, 855, 860, 863, 881, 884, 885		<code>\tabu@colX@rw</code>	90, 357, 358, 359, 365,
<code>\tabu@tabudecimal</code>	212, 1003, 1007, 1026		369, 417, 419, 429, 438, 442, 445, 446, 447
<code>\tabu@target</code>	69 , 138,	<code>\tabu@colXalign</code>	440
	228, 244, 252, 271, 337, 339, 340, 344,	<code>\tabu@decimal</code>	11 , 212, 1003
	350, 458, 560, 571, 575, 593, 597, 621,	<code>\tabu@defaulttarget</code>	
	624, 625, 627, 628, 631, 632, 637, 638,		31, 35, 195, 199, 340, 342, 343, 477
	644, 648, 658, 699, 1080, 1083, 1098, 1099	<code>\tabu@DisableCommands</code>	487
<code>\tabu@temp</code>	144, 146, 150, 153,	<code>\tabular</code>	177
	245, 247, 366, 370, 421, 425, 437, 445,	<code>\tabu@linestyle</code>	9, 972
	447, 605, 607, 608, 720, 724, 750, 807,	<code>\tabu@phantomline</code>	7, 779
	811, 815, 820, 830, 953, 955, 956, 957,	<code>\tabu@saveX</code>	697
	958, 960, 961, 1008, 1011, 1012, 1028,	<code>\tabu@vline@arg</code>	806, 809
	1029, 1030, 1031, 1032, 1033, 1034, 1035,	<code>\tabu@vline@argi</code>	815, 818, 825
	1036, 1037, 1038, 1039, 1040, 1041, 1042,	<code>\tabu@vline@argii</code>	823, 826, 828
	1043, 1044, 1045, 1046, 1055, 1056, 1057	<code>\tabu@vline@argiii</code>	815, 829, 835
<code>\tabu@titles</code>	345, 677	<code>\tabu@vline@rewrite</code>	799
<code>\tabu@to</code>	237, 242	<code>\thempfn</code>	286, 296
<code>\tabu@tr@cing</code>	27, 28	<code>\toks</code>	86, 1065, 1066, 1067,
<code>\tabu@tracing</code>	26, 27		1069, 1071, 1073, 1077, 1085, 1087, 1090,
<code>\tabu@tracing@save</code>	25, 684		1092, 1107, 1200, 1203, 1204, 1210, 1211

<code>\tracingtabu</code>	15	<code>\vcenter</code>	81, 785, 793
<code>\tracingtabu_(debugshow_option)</code>	20	<code>\verbatim</code>	10, 192, 212
<code>\ttfamily</code>	999	<code>\vskip</code>	847
U		<code>\vspace</code>	786
<code>\unhbox</code>	514	<code>\vtop</code>	754
<code>\unkern</code>	1319	W	
<code>\unless</code>	1213	<code>\write</code>	166, 169, 173, 478
<code>\unskip</code>	998, 1324, 1411	X	
<code>\usetabu</code> ...	12, 95, 102, 114, 366, 1113, 1134	<code>\xleaders</code>	899, 937
<code>\usetabu_(new_column_type)</code>	95	Z	
V		<code>\zap@space</code>	967
<code>\value</code>	285, 294, 484		
<code>\vbox</code>	785		