

# The `tabulary` package\*

David Carlisle

2003/09/14

## 1 User Documentation

`\begin{tabulary}{\langle length \rangle}{\langle pream \rangle} ... \end{tabulary}`

The rather daft name may change in a later release but it is a pun on `tabularx`, which itself was a pun on `tabular*`...

These environments work pretty much like the standard `tabular` environment (or more correctly, the enhanced version from the `array` package) except that there are more possibilities for the column types.

**LCRJ** These new ‘uppercase’ column types are only activated in the `tabulary` environment. In order to make the total table width equal to  $\langle length \rangle$  the LCRJ columns are converted to `p` columns (with `\raggedright`, `\centering`, or `\raggedleft` or normal justification respectively applied). The width of these converted columns is proportional to the natural width of the longest entry in each column.

To stop very narrow columns being too ‘squeezed’ by this process any columns that are narrower than `\tymax` are set to their natural width. This length may be set with `\setlength` and is arbitrarily initialised to 10pt. (If you know that a column will be narrow, it may be preferable to use, say, `c` rather than `C` so that the `tabulary` mechanism is never invoked on that column.)

Similarly one very large entry can force its column to be too wide. So to prevent this, all columns with natural length greater than `\tymax` are set to the same width (with the proportion being taken as if the natural length was *equal* to `\tymax`). This is initially set to twice the text width..

Narrow `p` columns are sometimes quite hard to set, and so you may redefine the command `\tyformat` to be any declarations to make just after the `\centering` or `\ragged...` declaration. By default it redefines `\everypar` to insert a zero space at the start of every paragraph, so the first word may be hyphenated. (See `DogBook`).

As the environment makes a standard L<sup>A</sup>T<sub>E</sub>X box, it will be indented by the paragraph indent at the start of a paragraph, and so will not fit on a line if given argument `\textwidth` unless it is preceded by `\noindent` or is in a `center` environment or some other environment with zero paragraph indent.

---

\*This file has version number v0.8, last revised 2003/09/14.

## 2 Features

You can use `\multicolumn` but if the multicolumn text turns out to be longer than the final calculated widths of the columns that it spans, then the final table will be too wide.

`\verb` doesn't work. (except in restricted version as in `tabularx`)

The whole table is evaluated twice, so take care with some  $\text{\TeX}$  constructions that may have side effects like writing to files.

## 3 Options

The following package option is defined:

**debugshow** Causes a lot of stuff to appear on the terminal. I find this invaluable, you may find it less so.

# 4 Examples

With C columns			
1	the rain in spain falls mainly on the plain	(an @ expr.)	the rain in spain falls mainly on the plain the rain in spain falls mainly on the plain
a	b	(an @ expr.)	c
a	b b	(an @ expr.)	c c
a			
With J columns			
1	the rain in spain falls mainly on the plain	(an @ expr.)	the rain in spain falls mainly on the plain the rain in spain falls mainly on the plain
a	b	(an @ expr.)	c
a	b b	(an @ expr.)	c c
a			
With L, R and C columns, and a \multicolumn			
1	the rain in spain falls mainly on the plain	the rain in spain falls mainly on the plain the rain in spain falls mainly on the plain	and now for something completely different
x	some multicolumn text across columns 2–4		
a	b	c	d
a	b b	c c	d d
a			

The following examples attempt to show the effect of the `\tymax` and `\tymax` parameters. One should also perhaps note that `\tymax` refers to the total column width (including any inter-column space, rules etc) but `\tymax` just refers to the width of the column entry (like the argument to the standard `p` column).

\tymin=0pt

\tymax=\maxdimen

Note how the first column is ‘squeezed’. In fact it is in such a narrow column that even ‘a’ produces an overfull box warning!

a	b	c c c c c	d d d d d d d d d d d d d d d d d d
	b	c c c c c	d d d d d d d d d d d d d d d d d d
	b	c c c c c	d d d d d d d d d d d d d d d d d d
	b	c c c c c	d d d d d d d d d d d d d d d d d d
		c c c c c	d d d d d d d d d d d d d d d d d d
		c c	d d d d d d d d d d d d d d d d d d

\tymin=20pt

\tymax=\maxdimen

Here increase `\tymax` so that columns b and a are not so narrow. ‘a’ is set to its natural width, and ‘b’ is set to `\tymax`.

[illegible]

\tymin=20pt

\tymax=200pt

In the previous example, the large d column dominated the table, being a lot wider than the c column. By reducing `\tymax` can limit the width of column d producing more even column widths, but now producing an entry for d that is longer than that for c.

[illegible]