

TABULARIZE

MICHAEL SHARPE

Tabularize and its companion Tabularize +space are *AppleScript* macros designed for T_EXShop's Macros Menu¹. Both are intended to make it easier to work with tabular material in the L^AT_EX source. Select the block of rows of an existing table, then run the macro. In both cases, it will try to tabulate the material by aligning the & and \\ elements in the rows. Tabularize tries for the minimal such alignment, while Tabularize +space adds a space to each side of each & and a space to the left of \\, for enhanced readability. For example, if you select the rows

```
abcd&&123\\  
AB&4567\\
```

then Tabularize will replace those rows with

```
abcd&    &123\\  
AB  &4567\\
```

while Tabularize +space will produce

```
abcd &      & 123 \\  
AB   & 4567  \\
```

The macros can handle a number of special situations commonly encountered in tables:

- Material following an unescaped % is not lost—it is replaced at the end of the source line.
- Material following \\ such as comments, or , e.g., an instruction to allow additional space after the line, like [6pt], is preserved.
- The search for \\ on a line looks for the last uncommented occurrence and discards if it is followed by a } somewhere in the uncommented text. In this way, it will ignore every \\ used in a minipage or parbox or similar structure allowing line breaks.
- Escaped ampersands (\&) are not handled as column breaks but as regular text.
- The macros' first real action is to replace all runs of whitespace (space and tab characters) by a single space, with leading and trailing spaces stripped off. This allows the macro to be rerun as many times as needed without introducing unwanted new space.
- Lines without a \\ are handed just as L^AT_EX would handle them, with the next line continuing the current row. This is taken into account when figuring column widths. See the examples below for more information.
- Special attention is given to long cell entries, possibly involving macros, parboxes and other L^AT_EX constructs. See the next section.

Date: February 5, 2015.

¹Thanks to Nils Enevoldsen, whose suggested the need for these macros and provided valuable feedback on my early versions.

HANDLING TABLES WITH SOME LONG ENTRIES

This is a headache when trying to tabularize, and the result will be useless if it leads to columns running off the right side of the screen. The point of view I took when writing these macros is that table rows should be laid out along the line of the following exemplar, in which it is assumed that the table has six columns, with columns 2 and 5 possibly containing longer source.

```
col1 & col2
&col3 & col4 & col5
&col6\\
```

Note that the rows you wish to contain lengthy source must be the last entries in the source rows, with the next & positioned on the following line. \TeX will then typeset this as one row, assuming you have set up the tabular environment with 6 columns. (Note that if you omit the the & before col6, then col5 and col6 will appear in the same column, separated by a space.

When Tabularize is run on rows of this type, it will limit the space allocated to the second and third columns by allowing col2 and col5 to contribute a maximum of 1 space, with the remainder flowing over the right end. Here's an example:

```
\begin{table}[htbp]
\centering
\begin{tabular}{@{} cccccc @{}} % 6 columns, no header
\hline
Pigs & \parbox[b]{2in}{There is quite a bit of material in this little parbox}
& Horses & flies & \shortstack{Line 1\\Line 2\\Line 3}
& The end\\
\hline
\end{tabular}
\end{table}
```

which would be rendered by \TeX as:

	There is quite a bit of material			Line 1	
				Line 2	
Pigs	in this little parbox	Horses	flies	Line 3	The end

If Tabularize is run on the content lines, the result is

```
\hline
Pigs & \parbox[b]{2in}{There is quite a bit of material in this little parbox}
& Horses & flies & \shortstack{Line 1\\Line 2\\Line 3}
& The end\\
\hline
```

The column widths are now set to [width of "Pigs ", 1 (not width of parbox source), width of " Horses ", width of " flies ", 1, width of " The end"]

Notice that the \\ embedded in the \shortstack were not mistaken for line breaks in the table.

SOME EXAMPLES

EXAMPLE 1:

```
\hline
a      &      bcd      &      efg \\ % line 1
\hline
hij&  k\&      &  10\%      \\% line2
```

tabularizes to

```
\hline
a  & bcd & efg  \\ % line 1
\hline
hij& k\& & 10\%  \\ % line2
```

(Note that an innocuous space has been added between \\ and %. This addition happens at the outset of the script so that searching for \% doesn't pick up this unintended fragment.)

EXAMPLE 2:

```
aaa &% line 1
bb\\
cc
&ddd&e\\
```

tabularizes to

```
aaa &      % line 1
      bb  \\
cc
      &ddd&e\\
```

(Explanation: Column 1 has width the max of lengths of "aaa " and "cc", column 2 has width the max of 0 (space between & and % in first line) and width of "ddd", column 3 has width of "e".)

This would also work as expected if the final \\ were not present.

EXAMPLE 3:

```
\noalign{\smallskip}\hline\noalign{\smallskip}
AAA &
a

b
c &
e
```

tabularizes to

```
\noalign{\smallskip}\hline\noalign{\smallskip}
AAA &
      a

      b
      c &
      e
```

(Explanation: Line 1 has one column (no & at end) and has length > 1 so it contributes a maximum of 1 to the width of column 1, while "AAA " has length 4, so column 1 is 4 wide. Column 2 has 3 entries of widths 1, 1, 2, and so has width 2, and column 3 has width 1. \LaTeX will place "a b c " in column 2, and just "e" in column 3. Note that the blank line is handled correctly.)