`tbl.typ`: a `tbl`-like preprocessor for Typst and `tablex`

Version TK

Max Rees

2023

Contents

1. Introduction

Typst [1] is "a new markup-based typesetting system that is powerful and easy to learn." While Typst provides a built-in `table()` function, it does not currently support more advanced features such as row spans and column spans, fine-grain control of borders, or complex cell alignments. Pg Biel's `tablex` project [2] provides many of these features. However, it remains the case that writing a table using either `table()` or `tablex()` can require rather verbose syntax.

The `tbl.typ` project is an effort to allow the expression of rich tables in Typst using a more terse syntax. This syntax comes from a UNIX heritage: the `tbl` preprocessor which designed for use with the traditional TROFF typesetting system [3] [4] [5]. Important differences between the syntax of traditional `tbl` and `tbl.typ` are noted later in this document.

After importing the library using `#import "tbl.typ"`, the basic format of a table when using `tbl.typ` is the following:

````
```tbl
Format specifications .
Data
```
````

The two main components of this syntax are:

- *Format specifications*. This describes the layout of the table in terms of the number and style of columns for each row.

  The last line of the format specifications must end in a period (`.`). This is the separator between the two sections.

- *Data*. This is the content that will fill each cell of the table. Generally every line of input in this section corresponds to a row in the table, though there are exceptions noted later. Cells are separated by the `tab` option which defaults to a TAB character.

2. Region options

In addition to the overall table syntax itself, you may specify *region options* that control the parsing and styling of the table as a whole using a "show-everything" rule prior to the tables you would like to control. For example:

```
#show: tbl.template.with(
  allbox: true,
  tab: "|",
)
```

The following options are recognized:

auto-lines, allbox	Like box, but also draw a line between every cell if true. This is the same option from tablex. *Default:* false
box, frame	If true, draw a line around the entire table. *Default:* false
breakable, nokeep	If true, the table can span multiple pages if necessary. *Default:* false
center, centre	Aliases for a tbl-align value of center.
decimalpoint	The string used to separate the integral part of a number from the fractional part. Used in N-classified columns. *Default:* "."
doublebox, doubleframe	Like box, but also draw a second line around the entire table if true. *Default:* false
font	The font for the table. Can be overridden later by the f(...) column modifier. *Default:* "Times"
header-rows	The number of rows at the beginning of the table to consider part of the "header" for the purposes of repeat-header. This option is also controlled by .TH rows in the table data. *Default:* 1

leading	The vertical spacing / leading to apply to table cells. Can be overridden later by the v(...) column modifier.
macros	A dictionary of (name, function) pairs that can be used with column modifier m(...). *Default:* (:)
pad	This is the padding used for each cell, for use with the Typst pad element function. The left and right keys can be overridden using a numeric column modifier. *Default:* (left: 0.75em, right: 0.75em, top: 3pt, bottom: 3pt)
repeat-header	If breakable is true and this option is true, then the table header controlled by header-rows will be re-displayed on each subsequent page. This option is also controlled by .TH rows in the table data. *Default:* false
stroke, linesize	How to draw all lines in the table. *Default:* 1pt
tab	The string delimiter that separates different cells within a given row of the table data. *Default:* "\t" (a TAB character)
tbl-align	How to align the table as a whole. *Default:* left

3. Format specifications

The format specifications section controls the layout and style of cells within rows and columns of the table.

Each comma or new line of format specification begins a new *row definition*. Within each row definition, encountering a *column classifier* character denotes a new column in the table. The classifier may be followed by any number of *column modifiers*, some of which may have required arguments enclosed in parentheses.

3.1. Column classifiers

The following column classifiers are recognized. They may be given as either capital or lowercase.

`L`	**L**eft align.
`R`	**R**ight align.
`C`	**C**enter align.
`N`	**N**umerically align.
`S`	This cell is column-spanned by the previous cell to the left in the current row. *The corresponding table data entries should be empty.*
`^` (caret)	This cell is row-spanned by the corresponding cell in the previous row above. *The corresponding table data entries should be empty.*
`_` (underscore), `-` (hyphen)	This cell contains a vertically-centered horizontal rule. *The corresponding table data entries should be empty.*
`=` (equals sign)	Same as `_`, but draw a double horizontal rule instead. *The corresponding table data entries should be empty.*
`|` (vertical bar)	This classifier does not actually begin a new column, but rather indicates the location of a vertical line. If placed at the beginning of a row definition, the line is drawn to the left of the first cell in that row. Otherwise, it is drawn to the right of the current cell in that row.

3.2. Column modifiers

The following column modifiers are recognized. They may be given as either capital or lowercase.

`b`	**B**old text using the Typst `strong` element function.
`d`	**D**own — set the vertical alignment to `bottom`.
`e`	**E**qualize the width of all columns with this modifier to the maximum width among those columns. This overrides modifier `x`.
`f(...)`	**F**ont name to use is given in parentheses. `f(B)` is an alias for the `b` modifier. `f(I)` is an alias for the `i` modifier. `f(BI)` is an alias for providing both of the above modifiers.
`i`	**I**talicize text using the Typst `emph` element function.
`m(...)`	**M**acro (function) to apply to each corresponding cell. The macros must be scoped using the `macros` region option. The macro currently only receives a single argument: the content of the cell. A future version may also pass the position of the cell in terms of row number and column number.
`o(...)`	Fill c**o**lor for the cell is given in parentheses.
`p(...)`	**P**oint size of the font is modified according to the argument in parentheses. If the argument begins with a `+` or `-`, then the argument is added or subtracted respectively with respect to the current size. The argument may be suffixed by a unit. If no unit is specified, `pt` is assumed. Valid units are: • `pt`, `p`: points. • `mm`: millimeters. • `cm`, `c`: centimeters. • `in`, `i`: inches. • `em`, `m`: `1em` corresponds to the current font size. • `en`, `n`: one *en* equals half of an em. • `P`: six *picas* equals one inch. • `M`: 100 of these equals one em.
`t`	**T**op — set the vertical alignment to `top`.

`u`	"Stagger" the affected cells so that they appear **between** the current row and the previous one above.
`v(...)`	**V**ertical spacing (leading) is modified according to the argument in parentheses. The length argument provided is in the same format as `p(...)`, with a default unit of `pt` and `+` / `-` relative adjustments supported.
`w(...)`	**W**idth of the column is guaranteed to be at least as big as the argument in parentheses, which acts as a *minimum width*. The length argument provided supports the same units as `p(...)`, with a default unit of `en`. However, relative adjustments are **not** supported. This overrides modifier `x`.
`x`	E**x**pand the width of the column to `1fr`, which will consume all of the remaining horizontal space on the page or in the current container. Applying this modifier to multiple columns will divide that remaining space evenly between them. This overrides modifiers `e` and `w(...)`.
`z`	The corresponding cell is treated as if it has **z**ero width for the purpose of determining the width of its column.
Number	A number given as a column modifier is interpreted as a `en` length which is used as a *column separation*. This is the distance that separates the end of the current cell's content from the beginning of the next cell's content. If there is a vertical line between the two cells, then it will appear centered on this separation distance. The default column separation is controlled by the sum of the `left` and `right` keys of the `pad` option. When not specified, this defaults to `0.75em + 0.75em`, which traditional TROFF calls `3n`.

4. Data

5. Differences from traditional `tbl`

- Region options must be specified using a "show-everything" rule; they cannot be provided within the `raw` block itself.
- The `tab` option may be a multi-character string.
- The `linesize` option is expected to be a Typst color, length, or stroke; a dimensionless number does not work.
- The alignment point of numerically-centered cells that are in the same column as left-centered or right-centered cells is always centered with respect to the column as a whole (as if the classifier was `C`), rather than with respect to the widest `L` or `R` entry.
- All column modifiers that expect an argument must provide that argument in parentheses.
- The `o(...)` column modifier is a `tbl.typ` extension.
- Nothing special needs to be done to use equations within table entries, though numerically-aligned columns may behave unexpectedly until the `delim` option is implemented.
- An empty entry in the table data must be given even if the cell is spanned or contains a horizontal line.
- `\Rx` table entries are not handled. Use the Typst `repeat` element function instead, though this does not work well at the moment without a fully-functioning `w(...)` column modifier (see Known issues).

6. Known issues

- The following region options are not currently supported:
  - `delim` (GH#1)
  - `expand` (GH#2)
  - `nospaces`
  - `nowarn`
- The `pad` region option currently requires the `left` and `right` keys to be explicitly provided. Unexpected behavior may occur if `x` is provided as a key. (GH#3)
- The following column classifiers are not currently supported:
  - `A` (**a**lphabetic)
  - `||` (double vertical line)
- The `w(...)` (minimum **w**idth) column modifier currently only sets the width of text blocks; it does not yet affect other table cells. (GH#5)
- `.T&` in the table data is not currently supported. (GH#4)
- Within text blocks, `.\"` comments are not removed, and other TROFF commands are not rejected. (GH#6)
- A table data row consisting of only `=` (double horizontal line) is not currently supported.

7. Examples

Example 1: adapted from [4]

````
```tbl
lz  s | rt
lt| cb| ^
^ | rz  s.
left||r
l|center|
|right
```
````

left		r
l	**center**	
		right

Example 2: adapted from [5, p. 41]

````
```tbl
c c c
l l ne .
Fact|Location|Statistic
Largest state|Alaska|591,004 sq. mi.
Smallest state|Rhode Island|1,212 sq. mi.
Longest river|Mississippi-Missouri|3,710 mi.
Highest mountain|Mount McKinley, AK|20,320 ft.
Lowest point|Death Valley, CA|-- 282 ft.
```
````

Fact	Location	Statistic
Largest state	Alaska	591,004 sq. mi.
Smallest state	Rhode Island	1,212 sq. mi.
Longest river	Mississippi-Missouri	3,710 mi.
Highest mountain	Mount McKinley, AK	20,320 ft.
Lowest point	Death Valley, CA	– 282 ft.

Example 3: adapted from [4]

````
```tbl
r| l
r  n.
software|version
_
AFL|2.39b
Mutt|1.8.0
Ruby|1.8.7.374
TeX Live|2015
```
````

software	version
AFL	2.39b
Mutt	1.8.0
Ruby	1.8.7.374
TeX Live	2015

Example 4: adapted from [5, p. 43]

````
```tbl
cf(Courier New) s s s
c | cs s
c | cs s
c |c|c|c
c |c|c|c
l |n |ne |ne.
Composition of Foods

_
Food|Percent by Weight
\^|_
\^|Protein|Fat|Carbo-
\^|\^|\^|hydrate

_
Apples|.4|.5|13.0
Halibut|18.4|5.2|...
Lima beans|7.5|.8|22.0
Milk|3.3|4.0|5.0
Mushrooms|3.5|.4|6.0
Rye bread|9.0|.6|52.7
```
````

Composition of Foods			
Food	Percent by Weight		
	Protein	Fat	Carbo-hydrate
Apples	.4	.5	13.0
Halibut	18.4	5.2	…
Lima beans	7.5	.8	22.0
Milk	3.3	4.0	5.0
Mushrooms	3.5	.4	6.0
Rye bread	9.0	.6	52.7

Example 5: adapted from [5, p. 42]

````
```tbl
c s s
c | c | c
l | l | ne .
Major New York Bridges

_
Bridge|Designer|Length

_
Brooklyn|J . A . Roebling|1595
Manhattan|G . Lindenthal|1470
Williamsburg|L . L . Buck|1600

_
Queensborough|Palmer &|1182
|Hornbostel

_
||1380
Triborough|O . H . Ammann|_
||383

_
Bronx Whitestone|O . H . Ammann|2300
Throgs Neck|O . H . Ammann|1800

_
George Washington|O . H . Ammann|3500
```
````

Major New York Bridges		
Bridge	Designer	Length
Brooklyn	J . A . Roebling	1595
Manhattan	G . Lindenthal	1470
Williamsburg	L . L . Buck	1600
Queensborough	Palmer & Hornbostel	1182
Triborough	O . H . Ammann	1380
		383
Bronx Whitestone	O . H . Ammann	2300
Throgs Neck	O . H . Ammann	1800
George Washington	O . H . Ammann	3500

Example 6: adapted from [4]

````
```tbl
rb c  lb
r  ci l.
r|center|l
ri|ce|le
right|c|left
```
````

r	center	**l**
ri	*ce*	le
right	*c*	left

Example 7: adapted from [3]

```tbl
Cf(BI) Cf(BI) Cf(B), C C Cu.
n|n*#sym.times;*n|difference
1|1
2|4|3
3|9|5
4|16|7
5|25|9
6|36|11
```

n	$n \times n$	**difference**
1	1	
2	4	3
3	9	5
4	16	7
5	25	9
6	36	11

Example 8: adapted from [5, p. 42]

```tbl
c c
np(-2) | n | .
|Stack
|_
1|46
|_
2|23
|_
3|15
|_
4|6.5
|_
5|2.1
|_
```

	Stack
1	46
2	23
3	15
4	6.5
5	2.1

Example 9: adapted from [5, p. 37]

```tbl
n.
13
4.2
26.4.12
26.4. 12
26.4 .12
abc
abc\&
43\&3.22
749.12
```

13
4.2
26.4.12
26.4. 12
26.4 .12
abc
abc
433.22
749.12

Example 10: adapted from [5, p. 41]

```tbl
c s s
c c c
n n ne .
AT&T Common Stock
Year|Price|Dividend
1984|15-20|\$1.20
5|19-25|1.20
6|21-28|1.20
7|20-36|1.20
8|24-30|1.20
9|29-37|.30\*
```

AT&T Common Stock		
Year	Price	Dividend
1984	15-20	$1.20
5	19-25	1.20
6	21-28	1.20
7	20-36	1.20
8	24-30	1.20
9	29-37	.30*

Example 11

```
```tbl
cbo(luma(85%))
co(luma(95%))  c.
Grade|Points
A|$ >= 510$
B|$ >= 450$
C|$ >= 390$
D|$ >= 330$
```
```

Grade	Points
A	≥ 510
B	≥ 450
C	≥ 390
D	≥ 330

Example 12: adapted from [5, p. 44]

```
```tbl
cf(I) s s
c cw(1in) cw(1in)
ltp(9) ltp(9) ltp(9).
New York Area Rocks
Era|Formation|Age (years)
Precambrian|Reading Prong|>1 billion
Paleozoic|Manhattan Prong|400 million
Mesozoic|T{
#set text(hyphenate: true, overhang: true)
Newark Basin, incl.
Stockton, Lockatong, and Brunswick
formations; also Watchungs
and Palisades.
T}|200 million
Cenozoic|Coastal Plain|T{
#set text(hyphenate: true, overhang: true)
#set par(justify: true)
On Long Island 30,000 years;
Cretaceous sediments redeposited
by recent glaciation.
T}
```
```

New York Area Rocks		
Era	Formation	Age (years)
Precambrian	Reading Prong	>1 billion
Paleozoic	Manhattan Prong	400 million
Mesozoic	Newark Basin, incl. Stockton, Lockatong, and Brunswick formations; also Watchungs and Palisades.	200 million
Cenozoic	Coastal Plain	On Long Island 30,000 years; Cretaceous sediments redeposited by recent glaciation.

Example 13: adapted from [4]

```
```tbl
le le7| lw(10).
The fourth line|_|line 1
of this column|=|line 2
determines|_|line 3
the column width.|T{
This text is too wide to fit into a column of width 17.
T}|line 4
T{
No break here.
T}||line 5
```
```

The fourth line		line 1
of this column		line 2
determines		line 3
the column width.	This text is too wide to fit into a column of width 17.	line 4
No break here.		line 5

Example 14: adapted from [5, p. 45]

````
```tbl
cb s s s s
cp(-2) s s s s
c | c | c | c | c
c | c | c | c | c
r2 | n2 | n2 | n2e | nbe.
Readability of Text
Line Width and Leading for 10-Point Type

_
Line : Set : 1-Point : 2-Point : 4-Point
Width : Solid : Leading : Leading : Leading
_
9 Pica : 93 : --6.0 : --5.3 : --7.1
14 Pica : 450 : --0.6 : --0.3 : --1.7
19 Pica : 5 : --5.1 : 0.0 : --2.0
31 Pica : 3 : --3.8 : --2.4 : --3.6
43 Pica : 5.1 : --90000.000 : --5.9 : --8.8
```
````

Readability of Text

Line Width and Leading for 10-Point Type

Line Width	Set Solid	1-Point Leading	2-Point Leading	4-Point Leading
9 Pica	93	–6.0	–5.3	**–7.1**
14 Pica	450	–0.6	–0.3	**–1.7**
19 Pica	5	–5.1	0.0	**–2.0**
31 Pica	3	–3.8	–2.4	**–3.6**
43 Pica	5.1	–90000.000	–5.9	**–8.8**

8. References

[1] https://typst.app/

[2] Pg Biel, "Typst-tablex." https://github.com/PgBiel/typst-tablex

[3] https://man7.org/linux/man-pages/man1/tbl.1.html

[4] https://man.openbsd.org/tbl.7

[5] L. L. Cherry, and M. E. Lesk, "Tbl – a program to format tables," in *Unix Res. System*, A. G. Hume, and M. D. McIlroy, Eds., vol. 2, 10th ed., Murray Hill, New Jersey 07974: Holt Rinehart & Winston, pp. 35–51. [Online]. Available: https://9p.io/10thEdMan/tbl.pdf