

Hints & Tricks

Glistenings

Peter Wilson

Not all that tempts your wand'ring eyes
And heedless hearts, is lawful prize;
Nor all, that glisters, gold.

Ode to a Favourite Cat
THOMAS GRAY

The aim of this column is to provide odd hints or small pieces of code that might help in solving a problem or two. I have learnt to my cost that the quickest way to get a correct answer to a question on the `comp.text.tex` (`ctt`) newsgroup is to give an incorrect answer. I hope that the ideas below

always work but as the column title implies, there might be some dross among the nuggets.

Corrections, suggestions, and contributions will always be welcome.

One of the less frequently asked questions on `ctt` is whether there is a package for typesetting forms, and the answer is ‘no’. Nevertheless forms are still typeset via \LaTeX .

Little boxes on the hillside, little boxes
made of ticky tacky.
Little boxes, little boxes, little boxes all
the same.

Little boxes

MALVINA REYNOLDS, 1961
(Popularised by Pete Seeger)

1 Tick boxes

One common component of a form is the tick box.

Can you produce a tick box? Yes No

Can I produce a tick box? Yes No

The empty boxes above were produced by the `\tickbox` macro defined below, and the `\xbox` command produced the checked box.

```
\newcommand*\tickbox{\fboxsep Opt%
  \framebox[\height]{\vphantom{M}}}}
\newcommand*\xbox{\fboxsep Opt%
  \framebox[\height]{\vphantom{M}$\times$}}
```

If you want a bit bigger box, like , you could use this larger `\Tickbox` instead of `\tickbox`.

```
\newcommand*\Tickbox{\framebox{\phantom{M}}}
```

The length `\fboxsep` is the space between the frame of a `\framebox` and its contents, so you can adjust the size of this kind of box by changing `\fboxsep` or by using a different phantom character.

Or you might prefer , from

```
\newcommand*\TickBox{\fboxsep Opt%
  \fbox{\rule{0em}{1em}\rule{1em}{0em}}}
```

This definition uses invisible rules (a `\rule` with zero height or width cannot be seen) to control the size of the tightly fitting `\fbox`. In this case, as the rules are the same length as the box is square.

2 Blanks, dashes and rules

Along the lines of checkboxes, another regular component of a form is _____.

The last sentence ended with `\hrulefill{}`. to give a rule that stretched to margin. You can use `\hrulefill` more than once in a line, such as in the next line where it is used twice.

Last name: _____ First: _____

Or, you may want to have a rule of a particular length. The rule in the following line is just long

enough for the word ‘something’ to be typeset:

Put _____ here.

Put something here.

The previous two lines were input as:

```
Put \underline{\phantom{something}} here. \\  
Put something here.
```

Here is a variety of rules and blanks. In each case I’ve shown the code of the interesting part of the sentence above the typeset result.

1. ... in `\rule{10mm}{0.4pt}` the ...
Fill in _____ the blank.
2. ... in `\hrulefill{}` the ...
Fill in _____ the blank.
3. ... in `\xfill[0.5ex]` the ...
Fill in _____ the blank.
4. ... in `\srule{something}` the ...
Fill in _____ the blank.
5. ... in `\srule[0.5ex]{something}` the ...
Fill in _____ the blank.
6. ... in `` the ...
Fill in _____ the blank.

The macros used above are part of \LaTeX with the exceptions of `\xfill` and `\srule` which are defined below.

The `\xfill[⟨len⟩]` macro is like `\hrulefill` in that it draws a rule in the available space but you can use the optional `⟨len⟩` length argument to raise or lower the rule with respect to the baseline.

```
\newcommand*\xfill[1][0pt]{%
  \cleaders
  \hbox to 1pt{\hss
    \raisebox{#1}{\rule{1.2pt}{0.4pt}}}%
  \hss}\hfill}
```

`\srule[⟨len⟩]{⟨text⟩}` draws a rule the same length as the `⟨text⟩`, but does not typeset the `⟨text⟩`. You can use the optional `⟨len⟩` argument to alter the height of the rule with respect to the baseline.

```
\newcommand*\srule[2][0pt]{%
  \setbox0\hbox{#2}%
  \rule[#1]{\wd0}{0.4pt}}
```

The `\rule` command has an optional argument, which is the amount to raise (lower) the rule from its normal position at the baseline. \LaTeX provides two dashes, the en-dash (–), input as `--` and the em-dash (—), input as `---`. En-dashes are used as the separator in a number range, like 2–4. Depending on your country’s typesetting tradition an en-dash or an em-dash may be used instead of a comma as a phrase separator in normal text. Longer dashes may be used to indicate that something is missing; a 2em dash (—) for missing letters in a word and a 3em dash (—) to indicate a missing word. You can use `\rule` as a basis for longer dashes; for instance

Customs Declaration		CD 44	
May be opened officially			
<input checked="" type="checkbox"/> Gift <input type="checkbox"/> Commercial sample <input type="checkbox"/> Documents <input type="checkbox"/> Other			
Quantity and detailed description of contents	Weight lb.	oz.	Value
Toy			15
Scarf			12
For commercial items only <i>If known, HS tariff number and country of origin of goods</i>	Total Weight		Total Value
I, the undersigned, whose name and address are given on the item, certify that the particulars given in this declaration are correct and that this item does not contain any dangerous article or articles prohibited by legislation or by postal or customs regulations.			
Date and sender's signature			
PS Form 1234, March 2004			

```
\newcommand*{\iiedash}{% 2em dash
  \rule[0.5ex]{2em}{0.4pt}}
\newcommand*{\iiidash}{% 3em dash
  \rule[0.5ex]{3em}{0.4pt}}
```

3 Forms

I have found that often the easiest way for me to define a form is to use the `picture` environment as this lets me place things just where I want them. Here is a possibly boring example for a customs declaration form; the real form is about 10% smaller than the example.

```
\newcommand{\form}{%
\setlength{\unitlength}{1mm}
\begin{picture}(79,80)
\sffamily \scriptsize \thicklines
\put(0,0){\line(1,0){80}}
\put(0,5){\line(1,0){80}}
\put(2,4){\makebox(0,0)[t1]{\normalsize PS Form
  \textbf{1234}, March 2004}}
\put(0,14){\line(1,0){80}}
\put(2,13){\makebox(0,0)[t1]{Date and sender's
  signature}}
\put(0,26){\line(1,0){80}}
\put(2,25){\makebox(0,0)[t1]{%
  \begin{minipage}{76mm}
  I, the undersigned, ... regulations
  \end{minipage}}}
\put(0,30){\line(1,0){48}}
\put(0,39){\line(1,0){80}}
\put(2,38){\makebox(0,0)[t1]{%
  \begin{minipage}{44mm}
  \textbf{For commercial items only} \\
  \textsl{If known,} ... \end{minipage}}}
```

```
\put(56,38){\makebox(0,0)[t]{Total Weight}}
\put(72,38){\makebox(0,0)[t]{Total Value}}
\put(0,53){\line(1,0){80}}
\put(2,52){\makebox(0,0)[t1]{%
  \begin{minipage}{40mm}
  \CONT \end{minipage}}}
\put(0,60){\line(1,0){80}}
\put(2,59){\makebox(0,0)[t1]{%
  \begin{minipage}{40mm}
  Quantity ... \end{minipage}}}
\put(49,59){\makebox(0,0)[t1]{%
  \begin{minipage}{14mm}
  \hfill Weight \hfill \mbox{} \\
  lb. \hfill oz. \end{minipage}}}
\put(72,58){\makebox(0,0)[t]{Value}}
\put(65,52){\makebox(0,0)[t1]{%
  \begin{minipage}{14mm}
  \CVAl \end{minipage}}}
\put(0,68){\line(1,0){80}}
\put(14,61){\makebox(0,0)[b1]{\DBX Documents}}
\put(14,66){\makebox(0,0)[t1]{\GBX Gift}}
\put(34,61){\makebox(0,0)[b1]{\OBX Other}}
\put(34,66){\makebox(0,0)[t1]{\CBX Commercial
  sample}}
\put(0,80){\line(1,0){80}}
\put(2,79){\makebox(0,0)[t1]{%
  \begin{minipage}{76mm}\normalsize
  \textbf{Customs Declaration} ...
  officially \end{minipage}}}
% vertical lines
\put(48,26){\line(0,1){34}}
\put(64,26){\line(0,1){34}}
\thinlines
\put(56,26){\line(0,1){9}}
\put(56,39){\line(0,1){14}}
\end{picture}
\setlength{\unitlength}{1pt}
}% end of \form
```

The variable parts of the form (i.e., the non-commercial answers) are represented by the upper-case commands, which have to be defined for any specific instance of the `\form`. For the example shown the code to complete and display it is:

```
\let\GBX\checkbox    \let\DBX\tickbox
\let\OBX\tickbox \let\CBX\tickbox
\newcommand{\CONT}{\normalsize\rmfamily
  Toy \ \ Scarf}
\newcommand{\CVAl}{\normalsize\rmfamily
  \centering 15 \ \ 12}

\begin{figure}
\centering
\form
\end{figure}
```

4 Letters

I received the following¹ from Michael Barr regarding string comparisons. Perhaps someone can help?

Following your column in the recent issue of *TUGboat* [1] I have a problem in string comparisons that I don't think is solvable. At least I couldn't. Suppose you want to decide if an argument ultimately expands to nothing. Or if two arguments have the same ultimate expansion. What I was finally led to was similar to your `\strcfstr` on page 340, [Ed. reproduced here]

```

\newif\ifsame
\newcommand{\strcfstr}[2]{%
  \samefalse
  \begingroup
    \def\1{#1}\def\2{#2}%
    \ifx\1\2\endgroup \sametrue
  \else \endgroup
\fi}

```

except with `\def` replaced by `\edef`. This worked until the day that the argument (I was actually testing for being empty, but the difficulty is the same) happened to be a matrix. At this point, I got a curious error message about misplaced &. It turns out that while you can put a matrix (or any alignment) into a `\def`, you cannot put one into an `\edef`. This is a built-in ‘feature’ (pronounced ‘bug’) of \TeX , one that will not be repaired and one that it is apparently impossible to overcome.

Michael Barr

References

- [1] Peter Wilson. Glistings. *TUGboat*, 22(4):339–341, December 2001.

◇ Peter Wilson
 18912 8th Ave. SW
 Normandy Park, WA 98166
 USA
herries.press@earthlink.net

¹ I have exercised editorial privilege on the original.