

There is no simple means to go around this problem (at least I did not find a simple way out) while preserving the feature devised by L. Lamport of separating the footnote marks from the footnote texts, as he explains on pages 99 and 156 of the L<sup>A</sup>T<sub>E</sub>X book.

### 1st solution

Explicitly place multiple footnote marks as math exponents in the table entries. Afterwards typeset the footnote texts using only `\footnotetext[(number)]{text}` with its optional argument that agrees with the exponents that were set.

### 2nd solution

- Redefine a new boolean variable, say `tablenote`:  
`% \tablenote is false by default`  
`\newif\iftablenote`
- Redefine `\table` so that it sets `\tablenotetrue`.
- Redefine the `\footnotemark` and `\@xfootnotemark` commands so they operate on the `mpfootnote` counter, instead of `footnote`, if `tablenote` is true.
- Tag all your table entries that required tagging with the same mark (except the first one, which is marked with the full `\footnote` command) with the `\footnotemark[(number)]` that makes use of its optional argument.

### 3rd solution

*Define a new environment.* Locally redefine `\c@footnote` and `\thefootnote` to be equivalent to `\c@mpfootnote` and `\thempfootnote` respectively, using `\let`.

I used the first two solutions. The second one is definitely better, but it requires that you know where you put your hands within the internal L<sup>A</sup>T<sub>E</sub>X macros. The third solution seems very simple.

Maybe someone has an even better solution?

Claudio BECCARI  
 Department of Electronics  
 Institute of Technology of Turin, Italy  
 Corso Duca degli Abruzzi 24  
 I10129 – TORINO, Italy  
 E-MAIL: beccari@itopoli.bitnet

◇ Jackie Damrau  
 SSC Laboratory  
 Mail Stop 1011  
 2550 Beckleymeade Avenue  
 Dallas, TX  
 email: damrau@ssc.vx1.ssc.gov

## Errata: “See also” indexing with Makeindex

Harold Thimbleby

In *TUGboat* 12, no. 2 (page 290) I gave the L<sup>A</sup>T<sub>E</sub>X definitions to enable an author to obtain ‘see also’ entries in their index. I am grateful to Professor John C. Slattery of Texas A&M University for pointing out that they did not work.

The following correction works for me (using *Textures* and L<sup>A</sup>T<sub>E</sub>X 2.09), but not for Slattery who is using a NeXT, though the same version of L<sup>A</sup>T<sub>E</sub>X:

```
\def\subsee#1#2{\em see also\} #1}
%       the #2 consumes a comma
\def\nosee#1{}
%       consume the page number
\def\seealso#1#2{\index
  {#1!zzzzz@string\subsee{#2}|nosee}}
```

The intention is, given the definitions as shown above, and supposing index entries for “Scheme” (`\index{Scheme}`) occur on pages 147 and 401, this is how `\seealso{Scheme}{LISP}` would end up in the index:

```
Scheme, 147, 401
      see also LISP
```

If you have the problems reported by Slattery, `\seealso` must be written out in full with you manually replacing the parameters #1 and #2 with what you want.

I made two errors in the original note: First, I published a fragment of L<sup>A</sup>T<sub>E</sub>X without testing it exactly as it appeared in print. The second error was conceptual. I naïvely forgot that a T<sub>E</sub>X definition is referentially opaque: I had assumed that given `\def\seealso{x}`, then `\seealso` can be written for `x` (with the exception of various cases where `x` contains things like `\futurelet`). In my case I had checked `x` but not the form `\seealso` that I used in the article. I had been fooled by the innocent appearance of `\index{argument}` — and I had not appreciated the L<sup>A</sup>T<sub>E</sub>X manual’s remark that `\index` should not appear inside another command’s argument, as it does here with `\def`.

I apologize for inconvenience caused, and I will look forward to any suggestions for a general solution. Is there any way for macros like L<sup>A</sup>T<sub>E</sub>X’s `\index` to detect when they are being used improperly?

◇ Harold Thimbleby  
 Stirling University  
 Stirling  
 Scotland, FK9 4LA