

(Review. Leban requires as input the number of disks and the first tower as a sequence of one-digit numbers. This means that creation of the initial tower is not part of the paper. Furthermore, Leban develops a set of Lisp-like functions for T_EX. Apparently Knuth's list macros, T_EXbook Appendix D.2, have been overlooked. Leban's small list processing system is an example of reinventing the wheel, with the concept of an active list separator absent. As a consequence the printing of the towers is done by recursion which is not necessary when using Knuth's list separator appropriately to 'execute' the list.)

[5] Pittman, J.E. (1988): Loopy.TeX. *TUGboat* 9, no. 3, 289–291.

[6] Salomon, D. (priv. comm.)

◊ Kees van der Laan
Hunzeweg 57, 9893PB
Garnwerd (Gr), The Netherlands
cgl@rug.nl

L^AT_EX

The L^AT_EX Column

Jackie Damrau

Earlier Column Revisited

Claudio Beccari from the Institute of Technology of Turin, Italy, sent some interesting addenda and an answer to a question that appeared in the L^AT_EX column in *TUGboat* 11, no. 4. Without further ado, here are Claudio Beccari's comments.

1. The minipage footnote counter is named `mpfootnote`, not `footnote`. If you want minipage footnotes numbered with lower case roman numerals, you can change the appearance of the minipage footnote counter with the following command:

```
\renewcommand{\thempfootnote}%
  {\roman{mpfootnote}}
```

In any case, the footnote marks are typeset as exponents in math mode so that the math italic font is used by default. If you want the regular roman font, or any other font, you have to request it explicitly:

```
\renewcommand{\thempfootnote}%
  {\rm\alph{mpfootnote}}
```

2. The double spacing scheme suggested by J. Colmenares does not work if the commands—`\single` and `\double`—are issued while in `\normalsize`. In fact, `\baselinestretch` operates only when the size is changed, but `\normalsize` does not actually change size if the size is already normal. In order to have these commands work, you need to type: `\small\single` or `\small\double`.

By doing this, the `\normalsize` command embedded in `\single` and in `\double` acts as it should with the new value of `\baselinestretch` becoming active.

A better command, that does not require changes to size in advance and operates also when in `\normalsize`, unfortunately requires the category recoding of `@`. The macro appears below:

```
\newcommand{\single}%
  {\leadingstretch{1.0}}
\newcommand{\double}%
  {\leadingstretch{1.5}}
\makeatletter
\newcommand{\leadingstretch}[1]%
  {\let\@tempa\currsize
  \let\@currsize\empty
  \def\baselinestretch{#1}\@tempa}
\makeatother
```

Footnotes: Problems and Solutions

Sometimes you need to attach footnotes to entries in tables. If you enclose a `tabular` environment within a `minipage` environment within a `table` environment, your footnotes float around together with your table.

But if you have to tag two or more table entries with the same footnote number (or letter) while you have one footnote text, you cannot use `\footnotemark[number]` several times in the table and `\footnotetext[number]{text}` after the `\end{tabular}` and before the `\end{minipage}` commands.

In fact if you do so, you will notice that the footnote text will be tagged with a number while the table entries are tagged with letters (default).

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^AT_EX 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^AT_EX 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^AT_EX 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^AT_EX 2.09), but not for Slattery who is using a NeXT, though the same version of L^AT_EX:

```
\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^AT_EX without testing it exactly as it appeared in print. The second error was conceptual. I naïvely forgot that a T_EX 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^AT_EX 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^AT_EX’s `\index` to detect when they are being used improperly?

◇ Harold Thimbleby
 Stirling University
 Stirling
 Scotland, FK9 4LA