

are not necessarily available electronically and reach out to those who are advancing through the ranks from novice to L^AT_EXpert.

One goal I try to achieve when printing a solution in this column is not to imply this is the only solution. There are always many ways to solve a problem; however, the trick is to find the solution which is understandable and useful to the user and not necessarily to the half-dozen experts to whom the user has NOT turned. Users do not always wish to have the most detailed/clever/astute/elegant solution. The *role* of the T_EX or L^AT_EX consultant in this case *is* to be the person who provides a solution which is adequate to the problem and which is most useful and educational to the user.

Another goal I try to strive for is to reach out to novice users and invite them to submit their questions. We all began somewhere—some with advanced knowledge in computer programming; some just beginning to learn programming; and some with no programming background at all. There is a vast pool of knowledge in the T_EX community that needs to be shared and we should never treat a beginner's question—no matter how many times asked—like “That has been answered before, why ask again?”.

We should strive to invite the novice user into our community in a friendly fashion. If someone repeats a question, we should provide them with some helpful suggestions and a possible place where relevant answers can be found (e.g., T_EXhax, TUGboat). In doing this, we all add enrichment to the T_EX community in our teaching, to our own knowledge, and we have gained one more contact in our everlasting networking scheme.

◇ Jackie Damrau
SSC Laboratory
Mail Stop 1011
2550 Beckleymeade Avenue
Dallas, TX
email: damrau@sscvt1.ssc.gov

A comment on the L^AT_EX column

Nico Poppelier

1 Double spacing

In the L^AT_EX column of TUGboat 1, no. 1(4) (1990), Jackie Damrau presented two macros for changing from single line-spacing to double line-spacing. Everyone who has ever tried this knows that the simpler macros do not do what the author, Josephine Colmenares, claims them to do. The correct answer was already given by Jackie in her column in TUGboat 1, no. 1(1) (1990), namely

```
\newcommand{\single}{%
  \renewcommand{\baselinestretch}{1}
  \large \normalsize}
\newcommand{\double}{%
  \renewcommand{\baselinestretch}{1.5}
  \large \normalsize}
```

I will try to explain why only this works under *all* circumstances.

In the comment in `lfonts.tex` (version of 10 April 1989) we can read¹

```
% A SIZE COMMAND is something like
% \normalsize that defines a type size.
% It is defined by the document style.
% However, \normalsize is handled
% somewhat differently because it is
% called so often--e.g., on every
% page by the output routine. The
% document style defines \@normalsize
% instead of \normalsize.
```

The command `\normalsize` checks whether the current size is `\normalsize`.

- If it is, it switches to the `\rm` font
- If it isn't, it switches the new size by means of a call to the kernel macro `\@setsize`

Somewhere else in `lfonts.tex` we find

```
% Each size command \SIZE
% executes the command
% \@setsize\SIZE{BASELINESKIP}
% \FONTSIZE\@FONTSIZE
% which does the following.
...
% 3. Sets \baselineskip to
% \baselinestretch * BASELINESKIP
...
```

In other words, to switch between single and double spacing you have to do the following:

1. you start by re-defining `\baselinestretch`

¹ The comment is reformatted to fit into the narrow columns.

2. you make a switch to an arbitrary, *but really different*, font size
3. you switch back to `\normalsize`.

2 Footnotes in a minipage

I should also like to answer the minipage footnote problem Jackie reported. The answer is given on pages 91 and 99 of the L^AT_EX user's guide and reference manual. Alternatively, you can delve into the comment of `latex.tex`—a bit more effort—but the answer remains the same.

In the comment in `latex.tex` (version of 13 June 1989) we can read

```
% \minipage :
% similar to parbox, except it also
% ...
% changes footnotes by redefining:
%   \@mpfn      == mpfootnote
%   \thempfn    == \thempfootnote
```

Inside a `minipage` environment L^AT_EX doesn't use the `footnote` counter, but the `mpfootnote` counter. And so a solution is

```
\renewcommand{\thempfootnote}
{\arabic{mpfootnote}}
```

An example to prove that this solution works. This table is formatted without any re-definitions.

```
\arabic  arabic numeralsa
\roman   roman numerals (lower case)b
\alph    lower-case letters (1-26)
\fnymbol symbols (1-9)c
```

^a Nice and simple.

^b I don't like 'mcmlxxviii'.

^c Old-fashioned.

This table is formatted with the re-definition I described above.

```
\arabic  arabic numerals1
\roman   roman numerals (lower case)2
\alph    lower-case letters (1-26)
\fnymbol symbols (1-9)3
```

¹ Nice and simple.

² I don't like 'mcmlxxviii'.

³ Old-fashioned.

◊ Nico Poppelier
Elsevier Science Publishers
Academic Publishing Division
R&D Department
Sara Burgerhartstraat 25
Amsterdam, The Netherlands
n.poppelier@elsevier.nl

L^AT_EX Tree Drawer

Glenn L. Swonk

Abstract

Today, many software systems are analyzed, designed and developed in a top-down hierarchical manner. This is especially apparent with Object Oriented Programming and Design. Diagramming a hierarchical relationship can be cumbersome in T_EX or L^AT_EX. This paper describes a MS-DOS tool that can be used to simplify the diagramming process by taking a simple input format and producing a L^AT_EX picture environment source which can be *input* in a L^AT_EX document.

Introduction

While trying to document a hierarchical directory structure, I manually placed the directory nodes' coordinates in a source file from calculations made from a rough sketch. Not only was this a tedious operation, it was prone to error and required many iterations before the exact result was achieved.

What I really wanted was a simple way specify the hierarchical relationship and an automated way to generate the output L^AT_EX file. Using the UNIX `find(1)` utility, it was easy to generate a list of files or directories that were to be plotted in a L^AT_EX picture environment. In the simplest case, `find(1)` can be used to generate a list of all files *under* a specified directory using the command `find <dirname> -print`.

What resulted from this problem is a MS-DOS tool which I call the L^AT_EX Tree Drawer (LTD). It takes input from a file in the form of a `find(1)` output and generates a L^AT_EX picture output that can be used as input to a L^AT_EX document. This paper describes the implementation and use of this tool.

Samples

To best illustrate what LTD can do, a few samples are in order.

Example 1 illustrates the simplest example — a single parent node and its two children. From the input file in figure 1, the resultant picture is shown in figure 2.

```
parent
parent/child1
parent/child2
```

Figure 1: Example 1 Input File