

A L^AT_EX fledgling struggles to take flight

Peter L. Flom

National Development and Research Institutes, Inc.
71 West 23rd St., 8th floor
New York, NY 10010

A little about this article

I work as a statistical consultant and data analyst at a nonprofit research company. I also work as an independent statistical consultant, mostly to graduate students in the social and behavioral sciences. I've done almost no computer programming. (I did have one course in computer programming, but it was so long ago that we used punch cards and waited a day or more for our programs to run on the mainframe that took up most of the basement; I also write some very simple programs in R.)

When I read the first issue of *The PracT_EX Journal*, I was thrilled. Finally, someone was writing a journal for beginners. So, I wrote a very enthusiastic 'Thank you' to the editor (Lance Carnes), and he wrote back, thanking me for the feedback, and asking me to write an article. I said OK. And here it is.

I'm writing with two groups in mind: Beginners, and people who write for beginners. I'd like to offer both groups some perspective from someone who is just a little way along the path. I'd like to let the true beginners know that it is possible to learn L^AT_EX; after only a few months of intermittent use, I can do a lot—I have written entire articles in L^AT_EX, some of them with quite complicated organizational structure and with fairly intimidating formulas; I've also started doing some presentations in L^AT_EX, using the Beamer package. If I can do it, you can too. I'd like to give the teachers the perspective of a recent beginner, so that their efforts can have maximum reward; when I consider that so many people contribute to L^AT_EX, often without any monetary reward, I imagine that those people would like to have their efforts help as many people as possible to use L^AT_EX easily and well. This article is in three sections:

1. Introduction
2. Some suggestions for teaching beginners
3. Some hints for beginners

Work on this project was supported by NIDA grant P30 DA11041; I'd like to thank the editor and the reviewers for their helpful comments and encouragement.

Editor's note: This article was originally published in *The PracT_EX Journal*, issue 2005-2, <http://tug.org/pracjourn>.

I hope, however, that both teachers and learners will read all the sections—the division of material is not rigid.

How I started using L^AT_EX

Long ago, I used Nota Bene. This was a very nice word processing program, designed for use by scholars. But no one I knew used it, so . . . I then became a dissatisfied user of Microsoft Word for years. But it came with my computer, everyone else used it, journal editors liked it, so, I used it. Then, at the recommendation of a friend and colleague, I started using WinEdt to write R files (R is a language and environment for statistical computing and graphics). It's great for that purpose, but I noticed that it kept mentioning L^AT_EX. I looked into it a little, but it looked really hard, so I didn't do much.

Then, I saw on the R-help list that someone was writing a book on R for beginners. I asked if he wanted some help from a beginner. He said he did; but the files were in L^AT_EX. He expressed amazement that I didn't use it. But it looked really hard, so I didn't do much.

Then I wrote a grant proposal that included a lot of formulas. A consultant on the grant did not have Word on his machine. He recommended L^AT_EX; but my co-investigator wanted Word files. So, I started looking more into L^AT_EX, and into programs to convert Word into L^AT_EX and vice versa. The deadline was looming, so I wrote that grant in Word (using Math Edit), and wrote files out as rtf files, which my consultant could read. Still, some formulas didn't print right; or printed differently on different computers; it was a mess. So, I resolved to learn L^AT_EX. I've been using it more and more over the last 6 months or so, and now really prefer it to Word, for virtually everything. Maybe after reading another issue of this journal, I'll prefer it for absolutely everything.

Some suggestions for teaching beginners

Ease of use

L^AT_EX looks hard. When I first saw a `.tex` file, I wondered how anyone could ever learn to write such

stuff. There are reasons for this: L^AT_EX was (naturally) written and extended by computer scientists (Donald Knuth for T_EX, Leslie Lamport for L^AT_EX, and many others), and that's probably why it looks like a programming language.¹ When you are really expert at something, it's hard to remember what it was like to not be expert; when you are really talented at something, it's hard to empathize with the less talented — this is not to criticize the people who write for beginners, it's just the way people are.

Well, I am neither experienced nor talented at programming, so I *can* empathize; even moderately complex L^AT_EX files look indecipherable to true beginners (at least, they did to me). Part of this is due to how people are first exposed to L^AT_EX. The first `.tex` file I saw was one which was going to be a book on a statistical programming language. I think that many people who start using L^AT_EX do so because of the limitations of Word or Word Perfect, or some other program. Thus, the first things we want to write are complicated files. Also, for the people who write documentation, it's easy to get into tricky stuff quickly, and this makes sense — there's not much point in having pages and pages of very simple documents.

One way of making the learning curve a little less steep is to provide annotated programs. Another might be to provide more exercises and treat an introductory book more as a text.

So, if you're writing for true beginners, emphasize ease of use. And, as L^AT_EX becomes used by more people who are not and never were programmers, try to remember that we don't think the way you think. If you're a programmer who doesn't like statistics, maybe thinking about how you would like to learn statistics would help in how people like me like to learn things like L^AT_EX.

Distributions

Everything I see on L^AT_EX mentions several (or more than several) different distributions. This just confused the heck out of me. Is there a difference? (I still don't know.) Is one better than the other? (I still don't know.) Some are free, some are commercial — what advantages do the commercial programs have? (They must have *some* or the companies would go out of business.) I've heard about LyX, which is a WYSIWYG version of L^AT_EX — this seems nice, but what are the drawbacks? I wound up using T_EX Live, more or less by chance. Now I use proT_EXt, because that's what I got sent as a mem-

¹ Reviewers pointed out that most all document markup languages developed in the pre-GUI (graphical user interface) era looked like this.

ber of the T_EX Users Group (TUG). It would be good if some documentation could list the various distributions and what their strengths are, or state that there are no real differences.

Writing in L^AT_EX is not like writing in Word

In Word (and probably in other word processors) when you don't get what you want, it's often because the program is illogical. It does some things automatically, some (most?) of which make no sense. In L^AT_EX, though, when you don't get what you want it's often because *you* messed up. When I started writing things that were a little complex, I often got errors. This still happens. At first, this really annoyed me. It almost made me stop using L^AT_EX.

Then I realized I should look on this more like a programming problem: Debugging is often necessary, and this doesn't mean you're stupid. I got this from the minimal programming I've done in R, but others who have never done any programming at all may not get this attitude, and I didn't see it in any of what I've read. Programmers may be so used to this way of thinking that they don't think to mention it.

Adding packages

I find this very confusing.² I've read various help files on how to do this; I'm sure they're all correct, I know they're all written by experts. It seems to me, as a nonprogrammer, that they contradict each other. I know they really don't, because then they wouldn't all work. So, it must be that I am even more confused than I thought, which is saying something. I don't fully understand *why* this has to be so hard (as I said, I am no programmer).

The other free software I use a lot is R, which also runs on lots of platforms, and also has lots of additional packages written by lots of different people, but there, when you add a package, it does all the background work for you; you just find the package you want, click on it, and you're done. If it can't be made easy, then I would strongly urge recommending that beginners install everything — all the available packages — at once. Disk space is cheap, writing the files takes a while, but it only needs to be done once. That's what I wound up doing (by uninstalling all the files, and then reinstalling everything I could get all at once) and this worked perfectly.

To a large extent, these problems have been solved by proT_EXt, which automates a lot of this. But, as far as I know, it is only for Windows, and

² According to one reviewer, this may not be as difficult as I think it is — there are, apparently, tools for doing this that I am unaware of; I am just writing about what I know.

thus L^AT_EX users using other systems may still have the type of question outlined above.

Annotated programs

All the books and other material on learning L^AT_EX include numerous examples of L^AT_EX files, which is good. One of the best ways of learning is by example. But one way to make these examples even more useful would be to include extensive annotations, either in the margins, in footnotes, or in text immediately below the program. What I have in mind is something like the way many editions of Shakespeare have notes explaining terms and references that are unfamiliar. The first few times a command is used, it would be useful to include a note. Kopka and Daly [3] do a nice job of this in their “Sample L^AT_EX file” on pages 16–19; I’d like to see more examples like this.

Debugging and error messages

Whenever I do anything complicated in L^AT_EX (and sometimes when I do something simple) I get errors. The messages accompanying these are sometimes helpful, but often rather obscure, at least to non-programmers such as myself. It would be great to have a source that explains some of these error messages in ordinary English. It would also be great to have some reference on debugging.³

Some hints for beginners

L^AT_EX has to be *learned*

Word is designed not to be learned. It’s supposed to function right out of the box (whether it does or not is another matter); if you are used to Word, then you may think that you should be able to use L^AT_EX right out of the box. Well, maybe some people can. I couldn’t. On the other hand, as you learn L^AT_EX, you get more and more control over how your document looks.

Some resources

There are a *lot* of free resources available for L^AT_EX (see the CTAN website). A lot of these are wonderful, and some are intended for beginners. I know some people find these resources to be enough for them to use L^AT_EX very well. Personally, I like books. I keep three close at hand: *Math into L^AT_EX* [2] is on my desk, and *Guide to L^AT_EX* [3], and *The L^AT_EX Companion* [4] are on my bookshelf. I like books (as opposed to web-based material) in general because:

1. they have extensive tables of contents and indexes;
2. they are already bound and thus easy to flip through;
3. I am just old-fashioned enough to like being able to page through a book, and keep it open on my desk while I work on something complex.

I like the three books mentioned above for different things. *The L^AT_EX Companion* [4] is a great book, but not for beginners. It’s intimidating. It’s too big. It assumes knowledge. I think it should be the 3rd or 4th book a L^AT_EX user buys; it’s a great reference, but it still kind of intimidates me.

Kopka and Daly’s *Guide to L^AT_EX* [3] is the best introduction to L^AT_EX that I’ve seen. The book I use most now is George Grätzer’s *Math into L^AT_EX* [2] (it’s open on my desk as I write this, I just looked up how to type the author’s accented name). I use this all the time, partly because one of the main reasons I started using L^AT_EX was to typeset some complex mathematical formulas. All three of these books are very well organized and comprehensible, given their depth. Your taste in particular books may vary. Try out a few. Even if you buy a bunch of books before finding one or two you really like, it’s not that much money (after all, the software is free).

Another resource I find very helpful is the general mailing list for T_EX users, texhax@tug.org; for more information, see <http://tug.org/mailman/listinfo/texhax>.

Figure out what you need to know, and when you need to know it

L^AT_EX is huge. It does all kinds of things, plus a lot that I am sure I am unaware of. What you need from it depends on what kind of work you do. For instance, I need to do a lot with tables, equations, bibliographies and imported graphics; I had to learn these first. But I don’t have as much need to make my own drawings—I’ll wait. Learning about some different fonts would be fun, but not urgent (for *me*—this may be very urgent for *you*). I will probably never learn to typeset Sanskrit or musical notation. But just figuring out what is available can be a challenge.

One thing to do, after you can write basic documents, is to browse through various sources, including books and the CTAN website; Jim Hefferon wrote a good introduction to the website in the first issue of *The PracT_EX Journal* [1]. Try to follow the discussions on the mailing list. *TPJ* is very helpful; and then there’s *TUGboat*, which also contains more advanced material (sometimes, I don’t even understand the titles!).

³ I have since found that Kopka and Daly [3] do include a list of some error messages in an appendix.

Run files often

Run your file through L^AT_EX a lot. Each time you do something even a little interesting, where you have any doubt at all about whether what you are doing will work correctly, typeset the file. If you've only made one or a few changes since you last ran the file, then it will be easier to find your error. In the editor I use (WinEdt) you can also typeset a small part of your document (hit ctrl+shift+c). This saves a lot of time.

On a related note, make backups often, and give them names you will understand and remember later. In particular, if you've gotten something complicated to work reasonably well, but still want to tweak it a little, save the file that works before you forget how you got it to work. (For me, this happens most often with complex, multiline equations and with tables that have complicated structures.)

Look at examples

All the books I listed have *lots* of examples. Try to figure out how they work and how they could be changed. Fool around; see what happens.

Make a default preamble

As you learn more L^AT_EX, you will (probably) find that there are certain packages that you always want loaded. It's hard (at least for me) to remember which ones I want, so I made a default file;⁴ as of March 6, 2005, it looked like this:

```
\documentclass{article}
\usepackage{graphicx}
\usepackage{amsmath, amssymb, latexsym, amsthm}
\usepackage{exscale, mathrsfs}
\usepackage{caption2, float, chapterbib, natbib}
\usepackage[section]{placeins}
\usepackage{fancyhdr}
\usepackage{geometry}
\usepackage[symbol, perpage]{footmisc}

\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\theoremstyle{definition}
\newtheorem{definition}{Definition}

\begin{document}
\title{Put title here}
\author{Peter L. Flom}
\maketitle
  Sample text
\bibliographystyle{amsplain}
\bibliography{file name}
\end{document}
```

⁴ One of the reviewers commented that it would be better to make a .sty file; I, however, do not know how to do this.

Summary

As I get more and more used to L^AT_EX, I find it more and more useful. I am gradually using it for more and more documents. For me, the best things about using L^AT_EX, as opposed to Word, are

1. L^AT_EX directs my attention to things that need attention. It takes care of section formatting, typography, and so on; but it forces my attention to things like complicated mathematical formulas and complex tables.
2. The ability to typeset complex mathematical equations and know they will appear correctly on other people's computers and on printout.
3. The naturalness of section formatting (that is, with `\section` and related commands)
4. The ease of cross-referencing to different sections of a document (using `\label` and `\ref`).
5. The helpfulness of the L^AT_EX community in finding solutions.

The biggest barriers to using L^AT_EX are

1. Working with co-authors and editors who insist on Word files.
2. Formatting complex tables.
3. Learning to use my editor (WinEdt) more efficiently.
4. Remembering that getting an error message is not the computer telling me that I am stupid (careless, ignorant, forgetful . . . but not stupid).

I look forward to learning more, and to becoming more expert, and to finding ways to spread my L^AT_EX wings. Certainly writing this article helped me do so, I hope reading it helped you, as well.

References

- [1] Jim Hefferon. CTAN for starters. *The PracT_EX Journal*, 1, 2005. <http://tug.org/pracjourn/2005-1/hefferon>.
- [2] George Grätzer. *Math into L^AT_EX*. Birkhäuser, New York, third edition, 2000.
- [3] Helmut Kopka and Patrick W. Daly. *Guide to L^AT_EX*. Addison Wesley, Boston, fourth edition, 2004.
- [4] Frank Mittelbach and Michel Goossens. *The L^AT_EX Companion*. Addison Wesley, Boston, second edition, 2004.