

Cahiers **GUT** *enberg*

☞ LAMS- T_EX : A PUBLIC DOMAIN DOCUMENT
PREPARATION SYSTEM EXTENDED AMS- T_EX
☞ Michael SPIVAK

Cahiers GUTenberg, n° 12 (1991), p. 13-20.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1991__12_13_0>

© Association GUTenberg, 1991, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.

L \mathcal{A} M \mathcal{S} -T \mathcal{E} X: A Public Domain Document Preparation System

Extended \mathcal{A} M \mathcal{S} -T \mathcal{E} X

Michael SPIVAK

*The T \mathcal{E} Xplorators Corporation, 3701 W. Alabama, Suite 450-273
Houston, TX 77027 (U.S.A.) email: spivak@rice.edu*

Abstract. L \mathcal{A} M \mathcal{S} -T \mathcal{E} X provides three basic extensions to \mathcal{A} M \mathcal{S} -T \mathcal{E} X:

- (1) As the ‘L’ in the name implies, L \mathcal{A} M \mathcal{S} -T \mathcal{E} X provides the functionality of L \mathcal{A} T \mathcal{E} X, including (a) automatic numbering, together with symbol labelling and cross-referencing, for equation numbers, lists, chapter and section headings, figure captions, theorems, lemmas, etc., etc.; (b) automatic placement of floating figures; (c) automatic table of contents generation and tools for creating an index; (d) literal mode; and (e) bibliographies (including interfacing with BIB \mathcal{E} X, if desired). However the approach is rather different, with syntax that is generally much more concise, and *designed to provide the user with much greater flexibility*.
- (2) There are special macros, and extra fonts, for easily producing complicated commutative diagrams; the results are at least as good as those found in any professional books and journals. There are also special macros for partitioned matrices and “bordered matrices”.
- (3) Finally, extensive table macros provide all the special refinements expected from professional typesetters.

Keywords: \mathcal{A} M \mathcal{S} -T \mathcal{E} X, automatic numbering, commutative diagrams, cross-referencing, figure placement, L \mathcal{A} M \mathcal{S} -T \mathcal{E} X, tables.

1. What L \mathcal{A} M \mathcal{S} -T \mathcal{E} X is and isn’t

To avoid any confusion, it should be emphasized that L \mathcal{A} M \mathcal{S} -T \mathcal{E} X is *not* L \mathcal{A} T \mathcal{E} X and it is also *not* \mathcal{A} M \mathcal{S} -L \mathcal{A} T \mathcal{E} X (which is a sort of unsavory hybrid of \mathcal{A} M \mathcal{S} -T \mathcal{E} X and L \mathcal{A} T \mathcal{E} X). Instead, L \mathcal{A} M \mathcal{S} -T \mathcal{E} X is an extension of \mathcal{A} M \mathcal{S} -T \mathcal{E} X that provides the functionality of L \mathcal{A} T \mathcal{E} X, and a lot more besides. This talk is admittedly propaganda for L \mathcal{A} M \mathcal{S} -T \mathcal{E} X, so I hope I may be excused if I occasionally overemphasize the virtues of L \mathcal{A} M \mathcal{S} -T \mathcal{E} X, especially in comparison to L \mathcal{A} T \mathcal{E} X.

One such virtue is compatibility. L \mathcal{A} T \mathcal{E} X users often complain of the incompatibility of L \mathcal{A} T \mathcal{E} X and plain T \mathcal{E} X—a perfectly good plain T \mathcal{E} X document

often won't run through \LaTeX at all—and it appears that future versions of \LaTeX will be even much more incompatible. On the other hand, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ is very compatible with plain \TeX , and $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ extends this compatibility. In fact, there is even a special `pcompat.tex` file that allows one to avoid the slight incompatibilities that were originally introduced in $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ (though I feel that these particular incompatibilities have their advantages).

Special care was taken not to use the brackets [...] in any $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ constructions (since Scandinavian keyboards have letters instead of these keys) and to allow French styles that make various punctuation marks active characters. Another small virtue many will appreciate is that in $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ there are no “fragile” commands. In fact, in the files that $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ writes for the table of contents, index, etc., everything appears exactly as typed, without any expansion of control sequences (or of active characters like < and > used for guillemets).

2. How it started

I should explain that originally I had very little interest in the main aspect of \LaTeX —automatic numbering, together with symbolic labelling and cross-referencing. I suppose that is partly because I had spent most of my time writing macros for \TeX rather than using it for long documents (though I had written *The Joy of \TeX*). But I think the main, partly subconscious, reason is that every time I looked at \LaTeX I sensed that things weren't designed in a way that I found inviting, though I never bothered to pursue the matter.

In fact, when I began a big macro writing project two or three years ago, I just wanted to be able to produce complicated commutative diagrams, like

$$\begin{array}{ccccccc}
 H_q & \xrightarrow{\omega} & \pi_{q+1} & \xrightarrow{h_{q+1}} & H_{q+1} & & \\
 \downarrow p_q & \searrow a \cong & \downarrow & \searrow b \cong & \downarrow & \searrow c \cong & \\
 H'_q & \xrightarrow{\omega'} & \pi'_{q+1} & \xrightarrow{h'_{q+1}} & H'_{q+1} & & \\
 \downarrow p'_q & & \downarrow \eta & & \downarrow p_{q+1} & & \\
 \mathcal{H}_q & \xrightarrow{\Omega} & \Pi_{q+1} & \xrightarrow{H} & \mathcal{H}_{q+1} & & \\
 \downarrow p'_q & \searrow \alpha \cong & \downarrow & \searrow \beta \cong & \downarrow & \searrow \gamma \cong & \\
 \mathcal{H}'_q & \xrightarrow{\Omega'} & \Pi'_{q+1} & \xrightarrow{H'} & \mathcal{H}'_{q+1} & & \\
 & & & & \downarrow p'_{q+1} & &
 \end{array}$$

so that I could use T_EX for commutative diagrams in books published by my publishing company, Publish or Perish, Inc.

Well, doing that wasn't so easy. In addition to all the macro writing, I had to create five new fonts consisting of various types of arrow pieces (heads, half heads, loops, etc.) at many different angles—and then get some one who really understood MetaFont to make good versions. But in the end I could produce commutative diagrams that looked as good, and usually quite a bit better, than ones found in professional journals; moreover, the macros made the production of such diagrams very straightforward and automatic.

But by this time I had come to recognize how important automatic numbering, and symbolic labelling and cross-referencing were to some people (including some prospective authors!). So I figured that I'd throw that in too.

And then, when I looked at L^AT_EX again, I quickly decided that I had to do that part differently. And then I decided that I needed to do other parts differently, too. So the result is that now, two or three years later, I finally have a macro package that I really like.

3. Automatic numbering, symbolic labelling, and cross-referencing

L^AT_EX's automatic numbering scheme can probably be summed up as follows: it's great if you want to do things the way L^AT_EX wants to do things, not so great if you want to do things your way. L^AM_S-T_EX, by contrast, aims to be a friendly document preparation system, that does things automatically, but in the ways you prefer.

For example, I seldom number formulas consecutively in a book. Occasional important formulas might be tagged as (*) or (**) or (A), etc., and within a proof I might tag a few formulas (1), (2), (3), . . . , with numbers that are used to refer to them only within that proof.

More significantly, after formula (14), you might have three aligned formulas that should be numbered (15a), (15b), (15c). Or the next formula might be (14*) because it's some kind of dual to (14) [I'm not making that up, it occurs in a recent Cambridge Press book].

Similarly, I'd hate to think that the famous Lemmas A and B of the mathematician H. Cartan might have ended up being Lemmas 7 and 12, perhaps, just because of an unyielding automatic numbering scheme.

So I decided to have an integrated scheme for modifying all automatic

numbering. For example, in \LaTeX you give a formula

```
$$ . . . $$
```

a tag simply by typing

```
$$ . . . \tag {$$
```

which provides the tag number automatically. To supply a specific tag like (A) instead, one simply types this tag in double quote marks right after the `\tag`:

```
$$ . . . \tag"({\bf A})" $$
```

But you can also modify the automatic numbering itself. You can reset the `\tag` number for the next formula with

```
\Reset\tag{...}
```

and there is also

```
\Offset\tag{...}
```

to offset the tag numbering by a specified amount (we'll see what good this is in a moment).

Moreover, when \LaTeX prints a tag like '(10)', it is really printing

```
( pretag material 10 posttag material )
```

The 'pretag material' might be something like '3.2.' in section 2 of chapter 3: that would normally be determined by a style file. The 'posttag material' is normally empty, but can be changed with `\newpost`. For example

```
\Offset\tag0 \newpost\tag{*}
```

means that the next tag will have the same number as the previous one, but have a * after it. And if we

```
\define\taga{\newpost\tag{a} \tag}  
\define\tagb{\Offset\tag0 \newpost\tag{b} \tag}  
\define\tagc{\Offset\tag0 \newpost\tag{c} \tag}
```

then something like

```
$$\align
... & ... \taga \\
... & ... \tagb \\
... & ... \tagc
\endalign$$
```

will produce aligned formulas with tags like (15a), (15b), (15c).

`\Reset`, `\Offset`, `\newpost`, and similar constructions apply not only to `\tag`, but to any L_AT_EX construction that provides automatic numbering. There is also `\newnumstyle` to change the ‘numbering style’, from `\arabic` to `\alph` (a, b, c, ...) or `\Alph` (A, B, C, ...) or `\roman` (i, ii, iii, ...) or `\Roman` (I, II, III, ...). And there is even `\newstyle` to change the style in which numbers are formatted; these are most important for lists, where one might want something like a), b), c), ... , or i., ii., iii., rather than (1), (2), (3),

As in L_AT_EX, one can label various constructions with `\label{...}`, and then use `\ref{...}` to have L_AT_EX print the number that was automatically assigned. Moreover, there are variants of `\ref{...}` that allow you to obtain different aspects of the total label. For example, in addition to the total label, say ‘(3.2.14*)’, you can get the raw number, 14, or the combination ‘3.2.14*’ containing the pre- and post-material, but without the other stylistic aspects.

As an example of how useful this can be, consider a manual, like the L_AT_EX Manual, where sections are labelled as 3.1, 3.2, 3.3, ... , in Chapter 3. When referring to a section in that Chapter, I wanted to say simply ‘section 4’, not ‘section 3.4’, but I naturally needed the double number for references to sections in other Chapters. Since I could never be sure just which Chapter a section might end up in, I wrote a little macro, using the tools already provided by L_AT_EX, that automatically chose the single number for a section in the same Chapter, but the double number for a section in a different Chapter.

As another example of the flexibility afforded, I once labelled some material with a page label, and then referred to it at a later point, which happened to appear on the same page, with the embarrassing result that page 14 contained a reference to something ‘on page 14’. So I wrote a little macro that would print ‘above’ if it occurred on the current page, ‘on the previous page’ if it occurred on the previous page, and ‘on page *n*’ otherwise.

In addition to the various `\new...` constructions already mentioned, there is the somewhat different construction `\newword`, which applies only to certain constructions, namely those that produce certain words as part of their output. For example, in this paper the \LaTeX `\abstract` construction supplied the word ‘Abstract’ automatically. But I could have typed

```
\newword\abstract{...}
```

to change this word. This construction thus alleviates problems encountered when style files are used for other languages.

Finally, somewhat as an aside, I might mention that if you type

```
\fancyfootnotes
```

at the beginning of your document, then \LaTeX will make the footnote numbering on each page start with 1. There is a natural task for any two-pass system like \TeX or \LaTeX , and I can’t imagine why it hasn’t been done before (I personally just loath having footnotes numbered consecutively throughout a chapter or book).

4. Theorems, propositions, etc.

The automatic numbering of theorems, propositions, lemmas, corollaries, etc., involves numerous new complications, which can be illustrated with the sequence

LEMMA 6. *If n is odd, then n^2 is odd; and if n is even, then n^2 is even.*

COROLLARY 1 (CONVERSE). *If n^2 is odd, then n is odd; and if n^2 is even, then n is even.*

THEOREM 7 (PYTHAGORAS). *$\sqrt{2}$ is irrational.*

DEFINITION 8. An integer x is a *perfect square* if $x = y^2$ for some integer y .

? **CONJECTURE A.** *If x is not a perfect square, then \sqrt{x} is irrational.*

Thus, in this particular case, Theorems, Lemmas, and Definitions are all numbered together, but Corollaries begin at 1 after each Theorem or Lemma. In addition, Definitions are in `\rm` type. Finally, Conjectures are numbered completely independently, as A, B, C, ..., and they have a ? in the margin. Quite different arrangements might be desired by other authors or style files, of course.

Without going into the details, I'll just say that L_AM_S-T_EX makes it easy to create new control sequences, say `\Thm`, `\Lem`, `\Cor`, `\Def`, and `\Conj`, to produce the appropriate heading automatically, and provide the numbering, in any combination desired, including the provision that `\Cor` numbering should begin at 1 after each `\Thm` or `\Lem`.

Having `\Def` and `\Conj` formatted differently requires a bit more work, redefining a fundamental construction from `lamstex.tex`, but there are even special constructions to aid in this.

In general, I have tried to make L_AM_S-T_EX as “modular” as possible: stylistic modifications can often be made quite easily, with `lamstex.tex` taking care of all sorts of unpleasant details on its own.

5. Automatic placement of figures

I'll cover this topic briefly by saying that I sought to satisfy the requirements put forth in an article by David F. Rogers in Volume 9, No. 3 of TUGboat. Figures are normally placed at most two to the page, one on top, one on the bottom, at some point after the page where they are mentioned. However, you are also allowed to deviate from this scheme. For example, it is possible to allow more than two figures per page, to force a figure to the bottom of the page instead of allowing it to go on top, to produce figures right between two lines of a paragraph, etc., etc.

6. Tables

And for tables, let me merely say that all the standard requirements are there, including spanning heads, etc., together with many very special constructions. Most important, tables will not look professional unless struts are added to lines—without them, horizontal rules will look too close to the next line of text. L_AM_S-T_EX automatically adds struts to each line, and, of course, it is possible to modify these struts, either individually or on a permanent basis. Similar care is taken to insure that all aspects of truly professional looking tables can be produced. See also the table on the next page.

7. Conclusion

As you can tell from the accelerating pace of this paper, there really are too many L_AM_S-T_EX features to be covered in a short time. For example, I haven't even mentioned the extensive indexing facilities provided.

Time	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:40					
8:45-8:55					
9:00-9:10					
9:15-9:25	TeX Instruction				
9:30-9:40					
9:45-9:55					
10:00-10:40					
10:45-10:55					
11:00-11:10					
11:15-11:25					
11:30-11:40					
11:45-11:55					
12:00-12:25					
12:30-12:40					
12:45-12:55					
1:00-1:10					
1:15-1:25					
1:30-1:55	Lunch	Lunch	Lunch	Lunch	Lunch
2:00-2:10	Concert		Assembly	Concert	
2:15-2:25					
2:30-2:40					

I would simply like to end by emphasizing two things: $\mathcal{L}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ was carefully designed to allow great flexibility in all its automatic features. And, just as $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ contains many, many features (some quite esoteric), to allow one to produce almost any sort of formula formatting ever required, $\mathcal{L}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ has also been endowed with many, many features, to address virtually all the needs of a document preparation system.