

Book Reviews

Book review: *The L^AT_EX Companion, Second Edition*

Claudio Beccari

Frank Mittelbach, Michel Goossens, with Johannes Braams, David Carlisle, and Chris Rowley, with the contributions of Christine Detig and Joachim Schrod, *The L^AT_EX Companion, Second Edition*. Addison-Wesley 2004, pp. xxviii+1092, ISBN 0-201-36299-6, USD 59.99, CND 86.99, \approx EUR 50.00.

The second edition of the indispensable *L^AT_EX Companion* upgrades the first edition published approximately ten years ago. But “upgrades” is an understatement: the second edition is about two times as large as the first one, and two times as many pages implies that the material in this new edition contains much more than just a few references to new packages or a couple of extra examples.

Although the topics covered are essentially the same as those of the first edition, the fourteen chapters, three appendices, and indices are completely rewritten and rich with displayed and numbered examples that, with a clever programming decision, are set up in such a way as to be completely faithful to the material being shown. The CD-ROM attached to the book contains a slightly reduced version of the T_EX Live distribution with the full running collection of the displayed examples, so that every reader can check directly also the details that are omitted from the typed page.

The chapters, after a good introduction, cover in order:

- the structure of a L^AT_EX document,
- the basic formatting tools,
- how to specify the layout of the page,
- how to typeset tabular material in a professional way,
- how to master floats,
- a clear discussion on fonts and encodings,
- how to typeset higher mathematics,
- how to use L^AT_EX in a multilingual environment,
- how to produce and handle graphical material,
- the organization and the tools for the difficult task of generating one or more indices,
- how to manage citations and produce useful bibliographies with the powerful tools that come with any distribution of (L^A)T_EX,

- and lastly, the tools for documenting class and package files.

The appendices start with a large overview on the commands usable in the document preamble and the more elaborate setups for package and class files. They continue with a very good analysis of the \TeX , \LaTeX , and class or package messages that show up when some error or other problems take place; this appendix is precious and most of its material has never been published before. The last appendix refers itself to the CTAN archives, where most extension packages reside, and to the various unusual sites from which to fetch the other files.

I do not want to go into the details of every chapter or appendix; it would be too lengthy. At the same time, whoever has used the first edition of the *LaTeX Companion* knows very well that this kind of book must be used as a descriptive manual; for this purpose the authors have written every chapter or appendix to be as “standalone” as possible, with an abundance of cross references allowing the readers to more deeply explore particular topics.

I would like to comment on a couple of points that I found a little weak or not sufficiently treated. This is not meant to criticize such an excellent and indispensable book as this one, but for contributing a line of thought that may help the readers (as well as the authors if they write a third edition in another ten years . . .).

One point is the presence of “typos”, that is spelling errors; actually the book is virtually devoid of any real typo, but there are some “cut and paste” errors, an average of two or three per chapter. This shows the amount of attention and accuracy the authors dedicated to correct and revise their source files; I have never succeeded in writing my own books with such a small number of spelling and/or “cut and paste” errors, so I admire them and praise their skill. But this raises another point: how is it possible to match such a beautiful typesetting engine as \LaTeX with a suitable editor that is aware of “cut and paste” errors? I know (or imagine) the authors used *emacs*, the most powerful ASCII editor available to anyone willing to climb its steep learning curve, but even this powerful tool is not capable of giving even a modest warning in such instances. I know that spell checking is one thing and grammatical checking is another, totally different. But at the end one would like to produce beautiful books, as \TeX was designed for, and one cannot avoid the painful task of reading over and over the galley proofs, and at the end it becomes such a tiring activity that errors sneak in anyway or remain undetected.

The few errors that bothered me a little bit are those that appear in Appendix B, where the treatment of errors should be absolutely error free, otherwise a reader can’t understand where is the error.

In Appendix B a slight confusion arises concerning $\backslash\text{long}$ macro definitions; for example in the unnumbered example starting at the end of page 932 there is an apparent inconsistency where $\backslash\text{lvec}$ is defined with $\backslash\text{newcommand}$, that generates “long” macros, in contrast with the $\backslash\text{show}\backslash\text{lvec}$ command that displays a “short” macro. The explanation of this unusual behavior follows immediately after the $\backslash\text{show}\backslash\text{lvec}$ command output, but when the reader reaches that point it remains a puzzle from the output of the preceding unnumbered example on the same page, where the macro $\backslash\text{xvec}$ (defined without optional arguments) appears to be “long”.

A naive error dealing with the Greek fonts appears in table 9.10: capital Greek letters are never accented, except for the diaeresis on Iota and Upsilon; a capital initial of a lowercase word is preceded by its spirit and accent, so that Ὶ , for example, is nonsense in Greek; it should be Ὶ as a capital initial, or simply A within an all caps word.¹ A similar situation holds true for the diaeresis: iota and upsilon receive the diaeresis only when they *follow* another vowel with which they are not supposed to form a diphthong so that ῖ is nonsense, because the capital upsilon preceded by its diacritical signs plays the role of an initial capital and therefore it cannot be part of a would-be diphthong; in the middle of a word it should simply be ῖ .

In chapter 9 it would have been useful to cite the book on \LaTeX written by Apolostolos Syropoulos [2]; since the author is Greek he devoted a large part of his book to typesetting in languages different from English, and with non-Latin alphabets; he even illustrated the Mongolian script. . .

Another point I think is not sufficiently emphasized is the fulfillment of international standards, especially the ISO ones. I believe that international standards exist explicitly for helping people from all over the world to understand each other, at least on technical matters. I saw in the book the ISO standards mentioned in connection to the font encodings, and that’s good. But I did not see a word about the ISO standard where, for example, table 9.5 illustrates the alternative mathematical operators names for eastern European languages. Those names are specified by the ISO standards and any alternative

¹ Actually the authors do not specify which Greek font they used, but with the default *cbgreek* fonts it is quite difficult to produce the errors shown in table 9.10; in this text I had to cheat a little bit in order to reproduce those errors!

name is to be considered “illegal”; I understand the necessity of producing documents containing some text typeset in accordance with obsolete typesetting traditions, or books containing translations of the ISO (Latin abbreviations of the) mathematical operators, as Apostolos Syropoulos did for math documents devoted to young high school Greek students. But in such an eminent reference as the *L^AT_EX Companion* I’d have put a discouraging sentence for common everyday use of nonstandard names. I know the L^AT_EX special symbols, now replaced by the American Mathematical Society or Text companion fonts, contained the \mathcal{U} symbol; maybe when L^AT_EX was first introduced in 1985 in some countries that symbol was in common use, but the ISO standards deprecate it explicitly and now I rarely see it in new books; even in this case a discouraging sentence recalling the international rules would have been appreciated.

This topic could be extended also to the bibliographies, for example, where emphasis is given to the many tools for typesetting bibliographic references, with the aid of BIB_TE_X, and with the purpose of producing the fanciest citation schemes, without mentioning any of the ISO standards on the matter.

There is a third point I would like to comment on, namely the excessively discouraging sentences connected with the use of primitive T_EX commands. I agree that when one writes a new class or a new package the use of the powerful L^AT_EX commands should be preferred over the often obscure circumlocutions needed when using primitive T_EX commands, but there are some things that even the most powerful L^AT_EX commands cannot do; examples are the definition of macros with delimited arguments, and the conditional statements dealing with character and category codes, as well as those dealing with comparisons of control sequences.

For the former it is necessary to use the `\def` or the `\gdef` primitives, and for the latter it is necessary to use the `\if`, `\ifcat`, and `\ifx` primitive conditional commands. The whole L^AT_EX kernel contains such commands, some of them relics of the old L^AT_EX 209 kernel, but most of them are there simply because they are necessary and cannot be substituted with newer L^AT_EX commands (which, in any case, are eventually defined by means of those T_EX primitives). Of course it is dangerous to make definitions by means of the primitive commands, be-

cause there is the real danger of redefining essential internal commands, even if they are “protected” by the presence of the `@` character; when classes and packages are written that protection is not effective. Nevertheless it might have been a good idea to illustrate the right technique for checking the existence of the command to be (re)defined so as to avoid messing up the whole system; the `ifthen` package allows to check if a control sequence is undefined, but at the class or package level even the kernel macro `\@ifundefined` can be used.

There are other primitive T_EX structures that can be very useful in producing reliable code also for L^AT_EX, but I won’t insist on this point since it would carry me too far.

Let me reach the conclusion. The second edition of *The L^AT_EX Companion* is a must for every serious L^AT_EX user. It complements Lamport’s handbook, but it adds such a great amount of useful information and wealth of working examples that any user will find it invaluable. Moreover, these examples carry with them the authors’ great experience, and let’s recall that most of them are members of the L^AT_EX 3 team — who could know L^AT_EX better than they? The few errors or omissions are not really so essential, provided the user keeps a little warning light blinking in the background of his/her mind.

When ten more years have elapsed, perhaps we’ll see a third edition that will be even more useful, for the future L^AT_EX of the year 2014. I hope so, because ten years from now I’ll still be using L^AT_EX, for which I thank not only Leslie Lamport, who started the whole game, and the contributors of the many extension packages that enhance so much the basic typesetting interpreter and engine created by Donald Knuth, but also all the authors of this excellent book.

References

- [1] Mittelbach, Frank, Errata list for *The L^AT_EX Companion, Second Edition*, <http://www.latex-project.org/guides/tl2c2.err>.
- [2] Syropoulos A., *Digital typography using L^AT_EX*, Springer Verlag, New York, 2002.

◇ Claudio Beccari
Politecnico di Torino
Turin, Italy
beccari@polito.it