

L^AT_EX News

Issue 11, June 1999

Back in sync

The last release of L^AT_EX was delayed even longer than you have come to expect. We hope that it proved worth waiting for. It required a major integration of the code from several people and, independently, the introduction of the LPPL (see L^AT_EX News 10) plus several related changes to our internal systems. It therefore seemed sensible to wait until everything was complete rather than do things in too much hurry.

This seem to have been a successful strategy as the recent patch release was related to an isolated change that was done many months previously. If this release does not appear a lot closer to its nominal date then ... well, you will not be reading this sentence!

Yearly release cycles

With the year 2000 rapidly approaching, we intend to switch to a release frequency of just one per year (with patches if necessary) for the core of L^AT_EX 2_ε. These days the system is sufficiently stable that the original update policy is costing everybody more time than is now warranted.

LPPL update

Thanks to extensive and valuable input from Matt Swift (swift@alum.mit.edu) we now have a clearer and more detailed form of the L^AT_EX Project Public Licence. This release contains both the original version (in `lppl-1-0.txt`) and the updated version, LPPL 1.1.

The future of SliT_EX

We still get a very small trickle of reports about this part of the system (if you are no longer able to recall L^AT_EX 2.09 then you will know it as the `slides` class). We have not classified them (in our minds at least) as bugs since we have always known that there are many problems with this class. It is clear to us that the only sensible action would be to redesign the system completely; in particular, to remove much of its complexity whose purpose is to support 10-year-old overlay technology. However, this would take a lot too much time and would be completely out of proportion to its current usage.

We are therefore planning to make the `slides` class unsupported in the sense that any problem related to the use of invisible fonts is considered to be a feature (The L^AT_EX 2_ε manual by Leslie Lamport doesn't even

describe this part of the class any more). Of course, if it still has its enthusiasts then we are happy to cede it to their loving care (somewhat like a preserved steam locomotive, in some parts of the world).

Fontenc package peculiarities

The `\usepackage` interface normally ensures that a package is loaded only once. The `fontenc` package has become an exception to this rule: it can be loaded several times using different options, e.g., allowing the user to add a font encoding in the preamble. This comes at a price for package writers: the low-level commands (see `ltxclass.dtx`) used to check if a package was loaded, and with which options, do not work for the `fontenc` package.

New math font encodings

As we announced in L^AT_EX News 9, a joint working group of the T_EX Users Group and the L^AT_EX3 Project has developed a new 8-bit math font encoding for T_EX. The reason why this work is not yet released is because of other exciting developments in the world of math fonts and math characters. It is obviously wise to ensure that the encoding work is fully integrated with the available fonts.

Those interested are reminded that further information about the Math Font Group may be found on the World Wide Web at:
<http://www.tug.org/twg/mfg/>.

Tools distribution

The `multicol` package has now got a small but useful extension which allows you to force a column break where this is really necessary. This is done with the command `\columnbreak`, which can be used like `\pagebreak` (e.g., within paragraphs) except that it cannot have an optional argument and thus it always forces a new column.

Coming soon

Major work on a new class file structure to support exible designs is well under way; some of this work will be presented at the TUG'99 conference in Vancouver, Canada. With a bit of luck much of this work could be ready for integration into the next release—so watch this space!