

# L<sup>A</sup>T<sub>E</sub>X3 News

Issue 1, February 2009

## *Welcome to L<sup>A</sup>T<sub>E</sub>X3*

Momentum is again starting to build behind the L<sup>A</sup>T<sub>E</sub>X3 project. For the last few releases of T<sub>E</sub>X Live, the experimental programming foundation for L<sup>A</sup>T<sub>E</sub>X3 has been available under the name `expl3`. Despite large warnings that the code would probably change in the future, we wanted to show that there was progress being made, no matter how slowly. Since then, some people have looked at the code, provided feedback, and — most importantly — actually tried using it. Although it is yet early days, we believe that the ideas behind the code are sound and there are only ‘cosmetic improvements’ that need to be made before `expl3` is ready for the L<sup>A</sup>T<sub>E</sub>X package author masses.

## *What currently exists*

The current L<sup>A</sup>T<sub>E</sub>X3 code consists of two main branches: the `expl3` modules that define the underlying programming environment, and the ‘`xpackages`’, which are a suite of packages that are written with the `expl3` programming interface and provide some higher-level functionality for what will one day become L<sup>A</sup>T<sub>E</sub>X3 proper. Both `expl3` and parts of the `xpackages` are designed to be used *on top* of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, so new packages can take advantage of the new features while still allowing to be used alongside many of the vast number of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> packages on CTAN.

## *What’s happening now*

In preparation for a minor overhaul of the `expl3` code, we are writing a comprehensive test suite for each module. These tests allow us to make implementation changes and then test if the code still works as before. They are also highlighting any minor shortcomings or omissions in the code. As the tests are being written, our assumptions about what should be called what and the underlying naming conventions for the functions and datatypes are being questioned, challenged, and noted for further rumination.

At the time of writing, we are approximately half-way through writing the test suite. Once this task is complete, which we plan for the first half of 2009, we will be ready to make changes without worrying about breaking anything.

## *What’s happening soon*

So what do we want to change? The current `expl3` codebase has portions that date to the pre-L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> days, while other modules have been more recently conceived. It is quite apparent when reading through the sources that some unification and tidying up would improve the simplicity and consistency of the code. In many cases, such changes will mean nothing more than a tweak or a rename.

Beyond these minor changes, we are also re-thinking the exact notation behind the way functions are defined. There are currently a handful of different types of arguments that functions may be passed (from an untouched single token to a complete expansion of a token list) and we’re not entirely happy with how the original choices have evolved now that the system has grown somewhat. We have received good feedback from several people on ways that we could improve the argument syntax, and as part of the upcoming changes to the `expl3` packages we hope to address the problems that we currently perceive in the present syntax.

## *What’s happening later*

After the changes discussed above are finished, we will begin freezing the core interface of the `expl3` modules, and we hope that more package authors will be interested in using the new ideas to write their own code. While the core functions will then remain unchanged, more features and new modules will be added as L<sup>A</sup>T<sub>E</sub>X3 starts to grow.

Some new and/or experimental packages will be changing to use the `expl3` programming interface, including `breqn`, `mathtools`, `empheq`, `fontspec`, and `unicode-math`. (Which is one reason for the lack of progress in these latter two in recent times.) There will also be a version of the `siunitx` package written in `expl3`, in parallel to the current L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> version. These developments will provide improvements to everyday L<sup>A</sup>T<sub>E</sub>X users who haven’t even heard of the L<sup>A</sup>T<sub>E</sub>X3 Project.

Looking towards the long term, L<sup>A</sup>T<sub>E</sub>X3 as a document preparation system needs to be written almost from scratch. A high-level user syntax needs to be designed and scores of packages will be used as inspiration for the ‘out-of-the-box’ default document templates. L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> has stood up to the test of time — some fifteen years and still going strong — and it is now time to write a successor that will survive another score.

# L<sup>A</sup>T<sub>E</sub>X3 News

Issue 2, June 2009

## *T<sub>E</sub>X Live and the expl3 code*

T<sub>E</sub>X Live 2009 is almost upon us, and the L<sup>A</sup>T<sub>E</sub>X3 team have been readying a new release of the experimental L<sup>A</sup>T<sub>E</sub>X3 code for this. Very dramatic changes have occurred since the last public release of the code in T<sub>E</sub>X Live 2008; no backwards compatibility has been maintained (as warned in the beginning of the documentation) but we believe the changes made are all much for the better. Almost every single part of `expl3` has been scrutinized, resulting in a far more coherent code base.

The `expl3` code is now considered to be much more stable than it was before; a comprehensive test suite has been written that helps to ensure that we don't make any mistakes as we change things in the future. In the process of writing the test suite, many minor bugs were fixed; we recommend such test suites for all similar developmental projects! Some small underlying changes are still expected in the `expl3` code, but major, disruptive, changes aren't planned.

## *Planned updates*

Until now, the last update to CTAN of the `expl3` bundle was for T<sub>E</sub>X Live 2008. Now that work on the code is happening on a semi-steady basis, we plan to keep updates rolling out to CTAN more frequently. This will allow anyone who wishes to experiment with the new code to use the T<sub>E</sub>X Live or MiK<sub>T</sub>E<sub>X</sub> updaters to install a recent version without having to 'check out' the SVN repository and install the packages manually.

## *New members*

We didn't say anything about it in the last status update, but Joseph Wright and Will Robertson are now members of the L<sup>A</sup>T<sub>E</sub>X Team. They have been working fairly exclusively on the `expl3` code.

It's worth repeating that L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> is essentially frozen in order to prevent any backwards compatibility problems. As desirable as it is to benefit from the new features offered by new engines XeT<sub>E</sub>X and LuaT<sub>E</sub>X, we cannot risk the stability of production servers running older versions of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> which will inevitably end up processing documents written into the future.

L<sup>A</sup>T<sub>E</sub>X3 will not be inheriting the same restraints, so stay tuned.

## *Some specifics*

Morten Høgholm will be presenting the recent changes in much more detail at TUG 2009. Here are some quick specifics for those interested. New code written and broad changes made to the `expl3` modules:

**More logical function names** Many function names that were hold-outs from the T<sub>E</sub>X naming system have been changed to fit into the more logical scheme of `expl3`; e.g., `\def:Npn` and `\let:NN` are now `\cs_set:Npn` and `\cs_set_eq:NN`.

**Defining functions and conditionals** Much thought was put into new ways to define functions and conditionals with a minimum of code. See `\cs_set:Nn` and `\prg_set_conditional:Nnn`.

**Smart comparisons** Comparisons can be made much more easily now, with familiar notation such as `\intexpr_compare_p:n{ #1+3 != \l_tmpa_int }`.

**Data from variables** A new function argument specifier `V` has been added for extracting information from variables of different types, without needing to know the underlying variable structure. Some other tidy-ups on the argument specifiers offered, partially as a result of the addition of this new one.

**l3msg** New module to deal with communication between L<sup>A</sup>T<sub>E</sub>X3 code and the user (info messages, warnings, and errors), including message filtering partially inspired by the `silence` package.

## *The next six months*

Having overhauled the `expl3` code, we now plan to perform an analogous process with the foundations of the `xpackages`. These are the higher-level packages that will provide the basic needs such as control of the page layout and rich document-level interaction with the user. As the groundwork for this layer of the document processing matures, we will be able to start building more packages for a L<sup>A</sup>T<sub>E</sub>X3 kernel; these packages will also be usable on top of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and serve as broadly customisable templates for future document design.

As gaps in the functionality offered by `expl3` are found (in some cases, we know that they exist already), the programming layer will be extended to support our needs. In other cases, wrappers around T<sub>E</sub>X functions that can be more usefully handled at a higher level will be written.