LaTeX3 News

Issue 1, February 2009

Welcome to LaTeX3

Momentum is again starting to build behind the LaTeX3 project. For the last few releases of TeX Live, the experimental programming foundation for LaTeX3 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 LaTeX package author masses.

What currently exists

The current LaTeX3 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 LaTeX3 proper. Both expl3 and parts of the xpackages are designed to be used *on top* of LaTeX 2$_\varepsilon$, so new packages can take advantage of the new features while still allowing to be used alongside many of the vast number of LaTeX 2$_\varepsilon$ 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-LaTeX 2$_\varepsilon$ 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 LaTeX3 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 LaTeX 2$_\varepsilon$ version. These developments will provide improvements to everyday LaTeX users who haven't even heard of the LaTeX3 Project.

Looking towards the long term, LaTeX3 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. LaTeX 2$_\varepsilon$ 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.