

L^AT_EX News

Issue 6, June 2011

A key aim of releasing ‘stable’ L^AT_EX3 material to CTAN is to allow users to benefit from new ideas *now*, and also to raise the profile of usable L^AT_EX3 ideas. This is clearly being successful, with `xparse` being of particular utility to end users. This increase in interest has been particularly notable on the new TeX.SX Q&A site.

The L^AT_EX3 Team expands

Raising interest in L^AT_EX3 developments has inevitably led to feedback on cases where the code base has required attention. It has also attracted new programmers to using L^AT_EX3 ideas, some more than others! Bruno Le Floch has over the past few months made many useful contributions to L^AT_EX3, and we are very pleased that he has recently joined the L^AT_EX3 Project.

Bruno has taken a particular interest in improving the performance and reliability of the `expl3` language. This has already resulted in new implementations for the `prop` and `seq` data types. At the same time, he has identified and fixed several edge-case issues in core `expl3` macros.

The ‘Big Bang’

In parallel to Bruno’s improvements, Joseph Wright initiated a series of ‘Big Bang’ improvements to L^AT_EX3. The aim of the Big Bang was to address a number of long-standing issues with the L^AT_EX3 code base. Development has taken place over many years, with the status of some of the resulting code being less than clear, even to members of The L^AT_EX3 Project! At the same time, different conventions had been applied to different parts of the code, which made reading some of the code rather ‘interesting’. A key part of the Big Bang has been to address these issues, cleaning up the existing code and ensuring that the status of each part is clear.

The arrangement of L^AT_EX3 code is now the same in the development repository and on CTAN, and splits the code into three parts.

l3kernel The core of L^AT_EX3, code which is expected to be used in a L^AT_EX3 kernel in more or less the current form. Currently, this part is made up of the L^AT_EX3 programming layer, `expl3`.

l3packages L^AT_EX 2_ε packages making use of L^AT_EX3 concepts and with stable interfaces. The `xparse` and `xtemplate` packages are the core of this area. While many of the *ideas* explored here may eventually appear in a L^AT_EX3 kernel, the interfaces here are tied to L^AT_EX 2_ε.

l3experimental L^AT_EX 2_ε packages which explore more experimental L^AT_EX3 ideas, and which may see interface

changes as development continues. Over time, we expect code to move from this area to either `l3kernel` or `l3packages`, as appropriate.

In addition to these release areas, the development code also features a `l3trial` section for exploring code ideas. Code in `l3trial` may be used to improve or replace other parts of L^AT_EX3, or may simply be dropped!

As well as these improvements to the *code* used in L^AT_EX3, much of the documentation for `expl3` has been made more precise as part of the Big Bang. This means that `source3.pdf` is now rather longer than it was previously, but also should mean that many of the inaccuracies in earlier versions have been removed. Of course, we are very pleased to receive suggestions for further improvement.

L^AT_EX3 on GitHub

The core development repository for L^AT_EX3 is held in an SVN repository, which is publicly viewable *via* the Project website. However, this interface misses out on some of the ‘bells and whistles’ of newer code-hosting sites such as GitHub and BitBucket. We have therefore established a mirror of the master repository on GitHub.¹ This is kept in synchronisation with the main SVN repository by Will Robertson (or at least by his laptop!).

The GitHub mirror offers several useful features for people who wish to follow the L^AT_EX3 code changes. GitHub offers facilities such as highlighted differences and notification of changes. It also makes it possible for non-Team members to submit patches for L^AT_EX3 as ‘pull requests’ on GitHub.

As well as offering a convenient interface to the L^AT_EX3 code, the GitHub site also includes an issue database.² Given the very active nature of L^AT_EX3 development, and the transitory nature of many of the issues, this provides a better approach to tracking issues than the main L^AT_EX bug database.³ Developers and users are therefore asked to report any issues with L^AT_EX3 code *via* the GitHub database, rather than on the main Project homepage. Discussion on the LaTeX-L mailing list is also encouraged.

Next steps

The ‘Big Bang’ involves making a number of changes to `expl3` function names, and is likely to break at least some third-party code. As a result, the updates will not appear on the TeX Live 2011 DVD release, but will instead be added to TeX Live once regular updates restart (probably August).

Bruno is working on a significant overhaul of the `l3fp` floating-point unit for L^AT_EX3. He has developed an approach which allows expandable parsing of

¹<http://github.com/latex3/svn-mirror>

²<http://github.com/latex3/svn-mirror/issues>

³<http://www.latex-project.org/bugs.html>

floating-point expressions, which will eventually allow syntax such as

```
\fp_parse:n { 3 * 4 ( ln(5) + 1 ) }
```

This will result in some changes in the interface for floating-point numbers, but we feel that the long-term benefit is worth a small amount of recoding in other areas.

Joseph has completed documentation of the `xgalley` module, and this is currently being discussed. Joseph is hoping to move on to implement other more visible ideas based on the `xtemplate` concept over the next few months.

L^AT_EX News

Issue 7, February 2012

After the ‘Big Bang’

The last L^AT_EX3 News gave details of the ‘Big Bang’, in which the team have revised the layout and coverage of the L^AT_EX3 codebase. This process has made the status of different modules clearer, so that both the team themselves and everyone else know what is going on.

The ‘Big Bang’ changes were not shipped to CTAN until after the T_EX Live 2011 freeze, as we did not want to end up with a DVD containing badly broken code. The update went to CTAN soon after T_EX Live 2011 shipped, and has now propagated around the world. The new package naming (`l3kernel`, `l3packages` and `l3experimental`) has caused some surprises for a small number of users, but there have not been any major issues with the changes at the code level.

The ‘Big Bang’ has attracted attention from programmers outside of the L^AT_EX3 team, with useful feedback arriving on the LaTeX-L list and TeX.SX, in particular. One area that this has highlighted is the need to document carefully when changes to the ‘stable’ parts of the L^AT_EX3 codebase occur. All changes to `l3kernel` now come with an explicit date for the change in the documentation, which means that programmers can check exactly when the features they want were introduced.

Another key part of supporting L^AT_EX3 use beyond the team is making it easy to check on the version of L^AT_EX3 installed. To support that, the file date of the main `expl3` package is now set each time there is a release of the L^AT_EX3 material to CTAN. This means that the L^AT_EX 2_ε `\ifpackagelater` test can be used reliably to detect if the installed version of L^AT_EX3 is going to supply the functions that a programmer is using.

Deforming boxes

Additions to both the L^AT_EX3 stable material and more experimental modules continue. Joseph Wright has

been working on adding ‘native’ drivers for L^AT_EX3 to support box transformations. These allow box rotation, clipping and scaling with the drivers `dvips`, `xdvipdfmx` and direct PDF output.

The development of clipping support for the `xdvipdfmx` driver has also allowed us to suggest improvements to the L^AT_EX 2_ε graphics drivers, enabling clipping with the X_qL^AT_EX engine.

A TeX-based regex engine

Bruno Le Floch has been improving the efficiency and robustness of a number of L^AT_EX3 functions. Most notably, he has created a purely T_EX-based regular expression (regex) system for L^AT_EX3. This is currently experimental, but is already proving useful and will hopefully stabilise over the coming months.

Bruno’s regex system works with all of the supported engines (`pdfTEX`, `XqLATEX` and `LuaTEX`). He has implemented the core ideas of standard regex systems, along with some T_EX-specifics to allow matching and replacing the content of token lists by category code.

xparse improves

The `xparse` module has been overhauled, making the internal code more efficient and adding additional argument types. This has also allowed us to deal with a number of internal bugs, meaning that argument grabbing is now more reliable.

The argument grabbers themselves have been reworked so that in the event of an error, the user will normally get a meaningful message from T_EX rather than one pointing to `xparse` internal function names. This should help in tracking down erroneous input in real documents.

The galley

As detailed in the last issue, work on the galley module has been continuing. Discussion of Joseph’s reimplementation of the galley concepts highlighted some important areas to work on, with the nature of the template concept being particularly significant.

More work is still needed to finalise the galley concepts, but it is clear that some of this will require feedback from other areas. Joseph therefore hopes to finish work on the current round of galley improvements by the end of February, and to return to them once some other areas have been addressed.

Relationships between document items

The TUG 2011 meeting took place in October in India. Frank Mittelbach spoke there about ideas for describing the design relationship between document elements. These ideas allow a document designer to specify the design of a document element based on its context within a document, and progress in this area will likely lead to an extension in the `xtemplate` system.