# LaTeX3 News

Issue 3, January 2010

## *Happy New Year*

Welcome to the holiday season edition of 'news of our activities' for the LaTeX3 team.

## *Recent developments*

The last six months has seen two significant releases in the LaTeX3 code. In the CTAN repository for the xpackages,[1] you'll find two items of interest:

- A revised version of xparse; and
- The new package xtemplate, a re-implementation of template with a new syntax.

Special thanks to Joseph Wright who handled the implementations above almost single-handedly (with lots of input and feedback from other members of the team and members of the LATEX-L mailing list).

These two packages are designed for the LaTeX package author who wishes to define document commands and designer interfaces in a high-level manner.

**xparse** This package allows complex document commands to be constructed with all sorts of optional arguments and flags. Think of how `\newcommand` allows you to create a command with a single optional argument and xparse is a generalisation of that idea.

**xtemplate** This package requires more explanation. Xtemplate is designed to separate the logical information in a document from its visual representation. 'Templates' are constructed to fulfil individual typesetting requirements for each set of arguments; to change the look of a certain part of a document, instantiations of templates can be swapped out for another without (a) having to change the markup of the source document, or (b) having to edit some internal LaTeX macro. LaTeX $2_\varepsilon$ packages, such as geometry or titlesec, already provide parameterized interfaces to specific document elements. For example, one may use titlesec to change the layout of a `\section`: one modifies its layout parameters via `\titleformat` and `\titlespacing`. In a way, such packages define a template for a specific document element and some manipulation commands to instantiate it. However, the moment the intended layout is not achievable with one package you are on your own: either you have to resort to low-level programming or find some other high-level package which, of course, comes with its own set of conventions and manipulation commands.

The xtemplate package can be thought of a generalization of such ideas. It provides a uniform interface for defining and managing templates for any kind of document element and most importantly provides a uniform interface for instantiating the layout.

Thus the designer activity of defining or modifying a document class is limited to selecting the document elements that should be provided by the class (e.g., `\chapter`, `\section` `\footnote`, lists, ...), selecting appropriate "named" templates for each of them, and instantiating these templates by specifying values for their layout parameters. If a desired layout can't be achieved with a given template a different template for the same document element can be selected.

Programming is only necessary if no suitable template for the intended layout is available. It is then that a LaTeX programmer has to build a new template that supports the layout requirements. Once this task is complete, the template may be added to the selection of templates that designers and users may choose from to define or adjust document layouts seamlessly.

This is a slight gloss over the complexities of the package itself, which you can read about in the documentation. We've tried to document xtemplate clearly but we'd love feedback on whether the ideas make sense to you.

As an addendum to the introduction of xtemplate, the older template package will be retired in the near future. To our knowledge there is only a single package on CTAN that uses template, namely xfrac, and members of the LaTeX team are in the process of switching this package over to xtemplate. If you have any private code that uses template, please let us know!

## *Upcoming plans*

Having announced the updated xparse and the new xtemplate, the next stage of development will revolve around using these two systems in the other components of the xpackages, feeding back our experience in practise with the original ideas behind the designs and evaluating if the packages are meeting our expectations.

[1] `http://mirror.ctan.org/tex-archive/macros/latex/contrib/xpackages/`

LaTeX3 News, and the LaTeX software, are brought to you by the LaTeX3 Project Team; Copyright 2010, all rights reserved.

*Packages to tackle*

**xhead** The first work will be to create a new xpackage (probably called xhead), for typesetting section headings and other document divisions. Section headings are one of the more complex areas to work with, so the work should stress xtemplate enough to know if its current design is sufficient for most needs. Nothing has been released yet, but we'll announce further developments on the LATEX-L mailing list[2] in the mean time.

**galley** We also need to give galley the same treatment as xparse and xtemplate have already had. That is, we have an older implementation (in fact two) that needs some work before we're ready to release it to CTAN. The galley package is used to place material into the vertical list while typesetting but before page breaks occur. Since it works at such a low level, it is important to solidify this package before writing higher level design templates.
An issue we have to face is that to achieve best results, galley cannot be used in concert with LaTeX $2_\varepsilon$ code. This could limit its usefulness, and we may decide that it's better to scale back the features we're attempting, to allow better interoperability for existing packages and documents. More work remains before we can decide between these options.

[2] For details, see `http://www.latex-project.org/code.html`