Peter Wilson's Herries Press packages

Will Robertson

Abstract

In September 2009 I became the maintainer of the majority of Peter Wilson's LaTeX packages. This short article describes how this came about and what the different packages are.

1 Who is Peter Wilson?

Frequent readers will be familiar with his name as an active member of the LaTeX community with ten *TUGboat* publications in the last eight years. He is most well known for his prodigious memoir class. He also gave the keynote address at TUG 2007.[1]

2 Introduction

Earlier this year I was writing a class file for a local conference and wrote to Peter about a minor feature request in his abstract package. He replied, quickly accepting the modifications I'd suggested, adding at the end 'Would you like to take over the package? I'm slowly retiring from LaTeX (age is calling) and trying to pass things off to others for support.'

Knowing of Peter's wide variety of packages on CTAN, it didn't make sense to me that each package should end up being maintained by whoever next sent a support email. Instead, I presumptuously offered to take on maintainership for the whole lot. After all, Peter is a well-regarded figure in the community and surely his packages don't receive very many bug reports? (As far as I know, this is indeed the case. Ask me again in a few months.) Peter seemed to like this idea and promptly sent me a complete list of his packages. I knew he was prolific, but I didn't quite realise the full extent of what I was taking on. Thirty-two packages ended up with my name in them, which involved a certain amount of tedium for both me updating the contact details for each and for the tireless CTAN administrators responsible for uploading the new versions.

In this report, I'll discuss what it means to be the 'maintainer' for a package and list the range and capabilities of Peter's packages.

3 Author versus maintainer

LaTeX itself and the majority of the third party contributed software for it are free and open source software, licensed under the 'LaTeX Project Public License'[2] (LPPL). The LPPL is similar to other well-known free software licences such as the BSD and Apache licences in that software may be freely distributed and modified, but access to the sources of derived versions is not required (unlike the 'copyleft' licences such as the GNU General Public License). The only restriction on redistribution is that the modified software must be clearly differentiable from the original by *users* of the code (not just people who read the source for fun). The easiest and most fail-safe way of doing this is simply by changing the name of the new package.

The LPPL also contains an interesting component that I have not seen in any other free software licences: the concept of an explicit 'maintainer' for the work who is theoretically responsible for keeping it up-to-date and for receiving bug reports. Usually the author of the software will be the maintainer of the work, but people change and move on and often lose interest in dealing with code they wrote long ago and no longer use. The LPPL formalises the process for new people to come along and adopt old code, especially 'orphaned' works for which the original authors can no longer be contacted.

This idea of explicit maintainership solves a real problem in the long term. In the LaTeX world, CTAN is the first port of call for contributed software; if it has not been uploaded there, it generally won't be available in TeX Live or MiKTeX. When package authors lose interest in their code and abandon their work, it is not clear how fixes or additions to their packages should be handled.

One can't simply upload patched versions of other people's code to CTAN, even if the original author is no longer around. If this were not the case, the CTAN team themselves would have to vet each new 'unofficial' update, in effect acting as *de facto* maintainers for all orphaned code — a preposterous idea considering the amount of work they already do, and certainly out of the question. Having explicit maintainers for the software they administer, the CTAN team can theoretically ensure that someone, somewhere, is responsible for each and every piece of software they (re-)distribute.

Peter himself was maintaining a number of packages for authors pre-dating his own involvement with LaTeX. This puts me in the dubious category of being a 'third generation LaTeX package maintainer'. (I take comfort in knowing that I'm not the only one.)

[1] "Between then and now — A meandering memoir", *TUGboat* 28:3, 2007, http://tug.org/TUGboat/Articles/tb28-3/tb90wilson.pdf, http://river-valley.tv/keynote-address-between-then-and-now-a-meandering-memoir/.

[2] http://www.latex-project.org/lppl/

4 The Herries Press packages

Peter's packages date from at least as far back as 1996 and fall into several rough groups:

- Replacements and better interfaces for functionality in the standard classes.
- Features to ease programming in LaTeX.
- New and assorted document features.

In the remainder of this section are brief descriptions of the packages I am now maintaining, concluding with a short list of Peter's works for which I am *not* responsible. Where other packages exist with similar functionality, I've listed them as well (to the best of my knowledge — no doubt I've forgotten some).

CTAN holds the definitive version of each package, of course. Rather than printing a link for each of these packages to their CTAN location, simply use this URL to access them:[3]

`http://tug.ctan.org/pkg/`⟨*package name*⟩

They are all included in recent (and not so recent) TeX distributions.

Development or pre-release versions of these packages are available at GitHub, where bugs and feature requests may be filed:

`http://wspr.github.com/herries-press/`

The adventurous may even wish to fork the code there in order to suggest code changes, which I will probably accept without too much question.

4.1 Standard class improvements

The standard LaTeX classes (article, book, report) are notoriously inflexible. You would like to change how the abstract appears, say? Then redefine the `abstract` environment. Same thing with figure captions, and document titles, and so on. Sooner or later, someone writes a package that provides a convenient user interface; here are those of Peter's.

abstract Easily customise the `abstract` environment for one- or two-column typesetting.

appendix Provides additional appendixing[4] capabilities.

ccaption Provides many features for customising and extending captions in floating and non-floating environments. See also the caption package.

romannum Change (any combination of) various document counters, such as captions, sections, equations, etc., to use roman numerals.

tocloft Easily customise the table of contents and other 'List of ... ' sections. See also the titletoc package.

titling Easily customise the document title produced with `\maketitle`.

tocbibind Add (perhaps with customizations) the table of contents, bibliography, index, etc., to the actual table of contents.

tocvsec2 Adjust the relationship between section headings and table of contents listing, and adjust automatic section numbering, mid-document.

For example, remove the number from a group of subsections and suppress their appearance in the table of contents without changing their markup.

4.2 LaTeX programming tools

LaTeX 2_ε's programming interface is often limited, offering little more in some areas than plain TeX's 'primitive' functionality. Peter's packages in this area tend to provide abstractions for specific tasks in LaTeX, useful for other class or package authors.

bez123 & multiply Draw generalised bezier curves in LaTeX, and multiply lengths without overflow. See also pict2e and the more ambitious drawing packages PSTricks and pgf/Ti*k*Z.

chngcntr Change the rules for the resetting of counters, such as numbering equations per-chapter or per-document.

chngpage and changepage Tools to locally change the size of the typesetting space and to detect robustly whether a page is even or odd.

N.B. that the two packages perform the same tasks, but changepage is interface-compatible with memoir and should be used for all new code that require these features. chngpage is an older version that is *incompatible* with memoir.

docmfp Extend doc to aid documentation of code in other programming languages. See also xdoc2 and (more recently) gmdoc.

ifmtarg Robust and expandable test for 'emptiness' of a macro argument.

makecmds LaTeX equivalent for `\def` with the syntax of `\newcommand` and `\newenvironment` (i.e., creates or overwrites the definition with equal abandon).

newfile Convenient interface to TeX's file reading and writing commands.

nextpage Extending the family of `\clearpage` commands. (E.g., `\cleartoevenpage`.)

printlen Print lengths of counters in specified units (as opposed to points, the TeX default).

stdclsdv Detect whether the class provides `\chapter`, and other sectional divisions.

[3] Having said this, each package name is hyperlinked if you're reading this electronically.

[4] I think Peter invented this word.

4.3 New and assorted features

The final section contains packages that provide document authors with features not offered by standard LaTeX classes or packages.

anonchap Makes `\chapter` typeset like `\section`. (E.g., for converting a book chapter into an article without changing the sectioning markup.)

booklet Typeset documents arranged on paper to be folded into booklets.

combine Combine multiple entire LaTeX documents into a single output file.

epigraph Add quotation-like material at the beginning/end of sections or chapters.

fonttable Visualise a font's glyph repertoire.

hanging Typeset paragraphs with hanging indents, and enable hanging punctuation using active characters.

For hanging punctuation, the microtype package for pdfTeX is recommended instead (although I may, in time, update hanging to work without active characters in XeTeX, using the latter's `\interchartoks` feature).

hyphenat Control hyphenation: turn it off entirely or allow the use of analphabetic symbols in hyphenated words.[5]

layouts Visualise the design of the page layout.

midpage An environment to vertically centre its contents in the text block.

needspace Reserve a certain amount of space on a page when you want to insert some material without breaking it over pages; if it cannot fit it will be forced it to the next page if necessary (ending the current page prematurely).

pagenote Typeset end notes per chapter or per document. See also the endnotes package.

verse Typeset verse material. See also poemscol.

vertbars Place vertical bars in the margin of paragraph text. Based on lineno, with the same caveats. See also the changebar package.

xtab Extensions and improvements to the package supertabular for multipage tables.

4.4 Classes and packages that I do not maintain

memoir Arguably Peter's single most influential contribution to LaTeX; memoir is a complete replacement for the standard document classes. It incorporates many of the packages mentioned above and contains a suite of new functionality, all with a consistent interface for creating new document designs. Now maintained by Lars Madsen. See also the KOMA-Script classes.

memdesign Originally the first half of the memoir manual. While this document is titled 'A Few Notes on Book Design', it's actually an excellent primer on typesetting and typography in general. Essential reading.

ledmac, ledpar, and ledarab For typesetting critical editions, based on plain TeX code 'edmac' and others. Now maintained by Vafa Khalighi.

expressg METAPOST package for drawing diagrams that consist of boxes, lines, and annotations.

iso and iso10303 LaTeX packages and classes for typesetting ISO standards. Possibly out of date with respect to the current typesetting standards.

isorot Rotate document elements and paragraph text. Perhaps I should maintain this one as well; it seems to be of more general interest than the other 'iso' packages.

5 Conclusion

I think it's important for members of open source communities to pass down their work as they start to retire from the field. Having a succession of maintainers allows bugs to be fixed and removes any confusion about how updates to their work should be named and distributed.

Would I continue to take on maintainership of yet more packages? Generally speaking, yes, provided the workload doesn't increase too much. Of course, we can't continue working with older and older packages indefinitely; at some stage new solutions to old problems will be created that supersedes the old work. (Cue my current involvement with the LaTeX3 Project.) In which case, old bugs in old code don't really need to be fixed.

But as long as Peter's packages continue to be useful (and most of them certainly are), I believe they deserve at least enough attention to keep them ticking along smoothly.

◇ Will Robertson
University of Adelaide, Australia
will dot robertson (at)
latex-project dot org

[5] Good spot for a hyphen, there, hey?