Talbot packages: An overview

Nicola L. C. Talbot

Abstract

This article briefly describes my packages that are available on CTAN and how they came about.

1 Introduction

Many of my packages end up on CTAN, the Comprehensive TEX Archive Network (`http://www.ctan.org`), and subsequently in TEX distributions such as TEX Live and MiKTEX. (My other packages are not on CTAN because they are either bespoke or still experimental.) Most of the packages are distributed under the "LATEX Public Project License" (`http://www.latex-project.org/lppl`).

2 General packages

This section describes packages for general use.

2.1 datetime

Changes the format of `\today` and provides commands for displaying the time.

The datetime package was the first package I wrote. It started out as an example in a LATEX course I was teaching in the late 1990s, but I later decided it might be useful to others so I uploaded it to CTAN.

The package provides the command

`\formatdate{`⟨*day*⟩`}{`⟨*month*⟩`}{`⟨*year*⟩`}`

that formats the given date according to the current date style. The command `\today` is redefined as `\formatdate\day\month\year` so they both display the dates in the same style. The default date style is the UK style or the style of the current language if babel has been loaded. The style can be changed via package options or declarations. You can also define your own custom date format.

An analogous command

`\formattime{`⟨*hour*⟩`}{`⟨*min*⟩`}{`⟨*sec*⟩`}`

formats the given time according to the current time style. The command `\currenttime` will display the current time according to the same format as `\formattime`. The style can be changed via package options or using `\settimeformat`. You can also define your own custom time format.

2.2 fmtcount

Display the value of a LATEX counter in a variety of formats.

When I first wrote the datetime package, I provided commands that convert a number or ordinal to a string so that it could be used with the textual date and time styles. I realised that people might want to use these commands but not want the rest of the datetime package so I decided to split them off into a separate package. While I was at it, I thought I may as well add some other formats, such as binary and hexadecimal.

2.3 glossaries

Create glossaries and lists of acronyms.

(This replaces the now obsolete glossary package.) With the glossaries package, you can define terms and acronyms which can then be used throughout the document. The default plural (appending an "s" to the singular) can be overridden and you can additionally specify a symbol or use the "user" keys to add other information.

To make a sorted list of the terms or acronyms to be displayed in the document, you need to use either `makeindex` or `xindy` to collate and sort the entries. Only those entries that have actually been used in the text will be displayed in the glossary or list of acronyms.

The glossaries bundle also provides the following packages:

- glossaries-accsupp: This package provides an interface between glossaries and Heiko Oberdiek's PDF accessibility support package accsupp.
- mfirstuc: This package provides `\makefirstuc`, which capitalizes the first letter of its argument, unless the first thing in its argument is a command with a non-empty argument, in which case it capitalizes the first letter of that command's argument. For instance, `\makefirstuc{abc}` produces 'Abc', `\makefirstuc{\ae{}bc}` produces 'Æbc', and `\makefirstuc{\emph{abc}}` produces '*Abc*'.

2.4 datatool

Tools to load and manipulate data.

(This replaces the obsolete csvtools package.) With the datatool package you can create databases, either via supplied commands or by loading an external CSV file. The original version of datatool was slow and inefficient, but Morten Høgholm suggested a much better internal representation of the database, which has significantly improved compilation time. The datatool bundle provides the following packages:

- datatool: This is the main package of the bundle. It can be used to create databases, iterate through a database, determine whether elements

Editor's note: The *TUGboat* editors requested this article in recognition of the many fine packages Nicola has created.

are integers, real numbers, strings or currency, and it provides commands that interface with the fp package to perform numerical operations on database elements (for example, computing the total, the arithmetic mean or standard deviation).

- datapie: This uses the pgf package to represent the contents of a database as a pie chart.
- dataplot: This uses the pgf package to plot the contents of a database.
- databar: This uses the pgf package to represent the contents of a database as a bar chart.
- databib: This works with BibTeX to convert a bibliography into a database. It can then be manipulated (for example, sorted according to a particular field) and displayed. Since you can apply filtering when iterating through a database, you can use it to, say, only display entries since a particular year, or only display entries that are journal articles.
- person: This provides support for displaying a person's name and pronoun in a document, thus avoiding the cumbersome use of "he/she" etc when performing tasks such as mail-merging.

2.5 flowfram

Create text frames for posters, brochures or magazines.

This package came about because I became frustrated trying to design technical posters for conferences. The layouts tended to require four columns with a figure or table spanning a couple of the columns, either at the top or at the bottom. I decided to adapt the technique used to format two-column text so that I could have an arbitrary number of columns with custom dimensions and locations so that the document text would flow from one column to the next just as it does in a regular two-column document. In addition I designed "static" and "dynamic" frames whose contents had to be set via a command or environment. This meant that I could position the figures and tables wherever I liked.

The same technique can be used to design the layout of brochures and magazines so I developed the code to make it easier to create a mini-table of contents that could be put in a separate frame alongside the chapter heading and thumbtabs which could be used to navigate through the printed document. TeX's `\parshape` and Donald Arseneau's shapepar package can be used to shape the static and dynamic frames so that the text can, say, go around an image or go in an L-shape to fit snugly around a shorter passage.

Determining the layout, particularly the parameters for `\parshape` and `\shapepar`, can be tricky so I made a Java GUI to assist. This is discussed in Section 4.

2.6 probsoln

Generate problem sheets and their solutions.

I wrote this package to help my husband, Gavin Cawley, generate assignment sheets. It underwent substantial modifications when Alain Schremmer asked for assistance with his documents at `http://freemathtexts.org`.

The idea is to have a database of questions (optionally with their solutions) and you can select particular questions, all questions or a random set. A package option or declaration can be used to hide or show questions or solutions so it's possible to either have each solution after the relevant question or have the solutions bundled together in another part of the document.

2.7 bibleref

Format bible citations.

I wrote this package in response to a query on `comp.text.tex`. It's designed to provide consistent formatting of references to parts of the Christian bible.

2.8 doipubmed

Special commands for use in bibliographies.

This package provides the commands `\doi`, `\pubmed` and `\citeurl` for use in bibliographies. Maarten Sneep and Heiko Oberdiek's doi package provides a more robust `\doi`.

2.9 quotmark

Consistent quote marks.

This package provides a means of ensuring consistent quote marks throughout your document. When I wrote it, I was unaware of Philipp Lehman's csquotes package.[1] The csquotes package provides more functionality than my quotmark package.

3 Development packages

The packages described here are provided for package and class developers.

3.1 makedtx

Perl script to help generate DTX and INS files.

Classes and packages are typically distributed as a DTX file with an accompanying INS file that's used

[1] I must have used the wrong terms in my keyword search.

to extract the code. However, when developing the code, it's a nuisance to have to edit the DTX file and then extract the STY or CLS files to test the modifications. I much prefer to edit the STY and CLS files directly, but this means that when I'm ready to distribute the new version I have to then bundle the code into a DTX file. I wrote this Perl script to do this for me.

3.2 xfor

Reimplements the LaTeX for-loop macro.

The glossaries and datatool packages need to use `\@for` to iterate through lists. However, quite often, I only need to search for an element that satisfies a particular condition, and it would be useful to terminate the loop when the element is found, akin to the `break` statement in languages such as C and Java. The xfor package redefines `\@for` so the loop can be terminated after the end of the current iteration.[2]

4 Jpgfdraw

Vector graphics application for LaTeX users (distributed under the GNU General Public License, `http://www.gnu.org/copyleft`).

My favourite drawing application has always been `!Draw`, which came with the Acorn Archimedes and Acorn RiscPC. Even after I moved to GNU/Linux, I still used the RiscPC to create images, which I then converted to Encapsulated PostScript. In the end, this became impracticable. Since I wanted to learn Java, I decided to write a Java application based on `!Draw`. While I was creating it, I was also considering writing an application that would work as a GUI for my flowfram package. Since both tasks required much of the same code, it seemed sensible to combine them into the same application. The code for the flowfram package required methods for computing the parameters for `\parshape` and `\shapepar` so I decided to also provide those as part of the set of LaTeX tools that come with the application. Briefly, you can use Jpgfdraw[3] to:

- Construct shapes using lines, moves and cubic Bezier segments;
- Edit shapes by changing the control points;
- Extract the parameters for TeX's `\parshape` command and for `\shapepar` (defined in the shapepar package);
- Construct frames for use with flowfram;
- Pictures can be saved in Jpgfdraw's native binary format (JDR) or native ASCII format (AJR) or can be exported in LaTeX and image formats:
  - a pgfpicture environment for use in LaTeX documents with the pgf package;
  - a single-page LaTeX document (including the picture code);
  - a LaTeX package based on flowfram;
  - a PNG image file;
  - an EPS file; or
  - an SVG image file;
- Incorporate text, text-along-a-path and bitmap images (for annotation and background effects);
- Alternative text may be specified for use when exporting to a LaTeX file (e.g. if the text contains symbols or if it should be set in maths mode);
- Mappings may be used to specify what LaTeX font declarations should be used when exporting to a LaTeX file (e.g. so you can map Chancery to `\fontfamily{pzc}\selectfont`).

Jpgfdraw is still in the beta stage. Occasionally it doesn't redraw some parts of the screen that need updating and it doesn't always update the bounding box for text areas. I hope to be able to fix these problems eventually. It also seems to have a problem when run on Windows Vista with Java6. This is difficult for me to test as I don't use Vista.

5 Conclusion

There have been times over the past couple of years when I considered giving up maintaining my packages as a result of ill-health and other commitments, and there were a couple of occasions when I was on the point of giving up, but then I received emails thanking me for the work I'd done and I changed my mind. I'm glad I didn't give up, although my responses to queries are somewhat slower than they used to be.

In my work as a production editor over the past year, I have developed a class that uses combine and hyperref to produce books, containing collections of articles, that can either be printed or made available on-line. It's my hope that at some point I'll be able to upload this to CTAN as well. The more I write classes and packages, the more I learn, and that's always satisfying.

◇ Nicola L. C. Talbot
University of East Anglia
Norwich, Norfolk
U.K.
`N.Talbot (at) uea dot ac dot uk`
`http://theoval.cmp.uea.ac.uk/~nlct/`

[2] In this respect, it differs from `break`, which immediately terminates the loop.

[3] I'm not very good at thinking of good names: it's a Java drawing package that creates pgf code.