Modular LaTeX Preamble

Rios, Mohamar

`mohamar.rios@gmail.com`

October 26, 2015

Contents

I do all of my LaTeX development using this modular preamble. I've never seen a similar technique described elsewhere but I've found it very useful. The idea comes from the way that configuration files are managed in some linux programs such as Grub and The X Window System. I do know it is possible to roll my own environments and turn this all into a custom document class but my style hasn't really solidified yet and this method lets me continue making changes to it easily.

Here I'm providing a slightly more detailed explanation of each of the packages in my preamble (there are short comments after each package in the file itself). I'm doing this because many of the packages are more geared towards typography and are as such not the type of thing you're likely to encounter unless you actually go looking for it.

The preamble itself is modular. Meaning it's split up into many smaller documents that handle something different. There is a section for each part of it.

1 00.main.tex

This document contains all of the main packages I use.

1.1 Some preliminary sanity checking packages

There are two optional packages that I call before everything else.

The **nag** package doesn't actually make any changes to your document. Instead all it does is tell LaTeX to throw warnings whenever it sees anything considered bad form or obsolete. Refer to the famous *l2tabuen* document for more information.

The current version of LaTeX is LaTeX 2_ε, but there has been a LaTeX 3 version in the works since the early 1990s. The project has stalled and there are many doubts about it ever being completed but in the meantime a series of bugfixes have been released as the package, **fixltx2e**. In really recent versions of LaTeX (as of this year) these bugfixes are already implemented and if you include it then you will get a harmless warning when you compile.

1.2 The Document class and typography packages

Here is the main set of packages that really affect the look and feel of the document.

I use the **KOMA-script** family of document classes. They are very similar to the default family of document classes except that they also provide by default a number of useful commands and environments that are more geared toward typography. There are four classes provided though I typically only use the article class, **scrartcl**. Here is a list of all the document classes it supports.

scrartcl The KOMA-Script article class.

scrreprt The KOMA-Script report class.

scrbook The KOMA-Script book class.

sclttr2 The KOMA-Script letter class.

The *letterpaper* option sets the paper size to the standard US size and the *DIV* option sets the margins (a larger number means smaller margins). The *DIV* option is typically used for books where a portion of the page is lost to cropping, binding, and gutter so it attempts to make the margins really large. I'm sort of abusing the feature here by making the margins undersized so that it gives me more space for indentation (this throws some harmless warnings).

One of the features KOMA-ScriptThis includes is the **addmargin** environment that lets you indent blocks of text (note this using this is considered bad LaTeXform since this sort of formatting should be handled by environments). Alternatively it is possible to only include parts of KOMA-script (as individual packages) within another class. For instance, if you wanted to use the **addmargin** environment in the default **article** document class then you would include the package **scrextend** within your document.

Note that I manually set the font for the document title so that it matches the default article class more closely using the command \setkomafont.

The **geometry** package makes the margins smaller so that the page appears "full size". I currently have it disabled in favor of the *DIV* option in KOMA-script.

The **microtype** package is difficult to explain but easy to understand once you see it. Essentially it makes use of new LaTeX typographical features that are aimed at improving kerning in a document. The only case where you may not want to use it is when presenting numbers in a table as it will try to line them up in a weird looking way [1].

1.3 Input encoding and math standards packages

There are two packages that I use to manage general standards.

The **isomath** package makes a few changes to the way math is rendered so that it conforms to the international standard, **ISO 80000-2**.

The **inputenc** package lets one set the input encoding to **UTF-8** so that special characters will work in the source code.

1.4 Font packages and commands

Next are fonts.

I use the **mathpazo** package which provides the *Palatino* (text and math) typeface because I find it a little nicer to read. It's also a little thicker than the default *computer modern* typeface which is good since *computer modern* was developed in an era before laser printing when text actually printed thicker on the page than it does now [2]. **Note:** It is recommended to use a slightly different linespread (the vertical space between lines) when using *Palatino*.

The next piece of code allows the use of the iconic blackboard bold fonts (eg. $\mathbb{R}, \mathbb{Z}, \dots$) in Palatino.

Finally the **mathrsfs** package provides a nice uppercase script font (eg. $\mathscr{P}, \mathscr{R}, \mathscr{S}, \dots$) which give you a nice alternative to calligraphic fonts (eg. $\mathcal{P}, \mathcal{R}, \mathcal{S}, \dots$) in case you run out of letters.

1.5 AMS math packages

Here's a series of AMS (American Mathematical Society) packages that most math students writing LaTeX will be familiar with.

These are **amsmath**, **amsthm**, and **amssymb**. They provide many commands, environments, and redefine many LaTeX commands so that they'll look nicer in a mathematical context (refer to [3]).

The **mathtools** package extends these by providing a few extra things such as a command that lets you define functions like $\|x\|$ so you don't have to deal with keeping track of crazy left and right brackets.

1.6 Functionality providing packages

Finally we get to some more useful packages that don't just affect appearance.

The **color** package lets you color text. Unfortunately this only works in text-mode. For a mathmode version refer to mathcolor in 20.commands.tex.

The **cancel** package provides a cancel function that lets you cross out stuff in math-mode (it's useful for stuff like counting arguments, eg. $a_1, \cancel{a_2}, a_3, \cancel{a_4}, \dots$). It also lets you write the cancellation as an arrow (eg. $\cancelto{0}{\frac{1}{n}}$) but as you can see it doesn't play nice with inline text.

The **esvect** package lets you write vectors with nice arrows displayed over them, (eg. $\vv{AB}$).

The **IEEEtrantools** package gives you a bunch of IEEE environments for typesetting equations nicely. The **IEEEeqnarray** in particular tends to be very useful (see examples on other document)

The **wrapfig** package provides an environment that lets you wrap text around figures. There are plenty of documentation and examples online, so I won't elaborate on this.

The **verbatim** package lets you comment out blocks of LaTeX code with the **comment** environment. To see an example look at this part of the source code of this document. It is useful for debugging code since it lets you comment out large swaths of text without

The **enumitem** package extends the **enumerate**, **itemize**, and **description** environments so that they have some extra functionality with regards to labeling.

The **array** package extends the **array** and **tabular** environments so that they have some more formatting functionality.

The **hyperref** package allows the use of hyperlinks within the document.

The **amsrefs** package gives a very simple LaTeX environment that interfaces with bibtex (this handles citations).

2 10.tikz.tex

This document contains only the tikz related packages.

In this case it only calls the **tikz** package itself and the **cd** library for commutative diagrams. Having this separate allows one to quickly switch out libraries or packages easily without messing with the rest of the preamble (for instance if needing to draw automata or parse trees).

3 20.commands.tex

This document provides a series of command definitions that are not directly provided by a package.

Anyone that writes a lot of LaTeX tends to accumulate many of these. In some circles they are referred to as in-house commands. Many of these I made but a few I obtained from other users on stackexchange or other sources. You are likely to produce your own commands, probably different from these. As such I'll only cover the few that I believe are particularly useful.

The command **contradiction** gives a contradiction symbol ⨳ which can be used to replace □ in the appropriate proofs. Though it's worth mentioning that there are many other popular contradiction symbols in use and no symbol has really become standard. I obtained this from stackexchange [4].

The command **ts** lets you use superscripts in text mode. For instance if you want to talk about the 1st of January. I obtained this from stackexchange [5].

The command **mathcolor** lets you use colors in math mode (eg. $a_1, a_2, a_3, a_4, \ldots$). I obtained this from stackexchange [6].

I also use two simple vertical blank space commands **aspace** and **qspace**. They really just call \vspace with a specific distance. I use these commands to create vertical blank spaces between a question and it's answer (\aspace) and between two different questions. I know this probably isn't the best way to do this as it would actually be better to write a "question" environment but it's very simple and seems to get the job done. **Note:** I haven't found a satisfactory way of dealing with multipart questions.

4 Extra packages

There are a few packages that you'll only want to appear inside your actual tex document.

The *easy-todo* package lets you create todo items very quickly with the \todo command **1.This is an example**. Then you can turn the display of the todo-notes on and off by simply switching the option on the package from *enable* to *disable*. This package also lets you output a clickable list of all the todo's in the document together with their page number like this:

ToDo

Currently showing a list of all the todos throws a harmless warning due a bit of bad code and there is a fix floating around but for some reason it hasn't been fixed upstream. Either way it's harmless so don't worry if you see it.

References

[1] Siarhei Khirevich (http://www.khirevich.com/), *Tips on Writing a Thesis in LaTeX*, Tex Stack Exchange. URL: `http://www.khirevich.com/latex/microtype/` (version: 2015-10-24).

[2] NauC (http://tex.stackexchange.com/users/15946/nauc), *Fatter Computer Modern*, Tex Stack Exchange. URL: `http://tex.stackexchange.com/a/101013` (version: 2013-03-05).

[3] Mico (http://tex.stackexchange.com/users/5001/mico), *What are the differences between ££, \[, align, equation and displaymath?*, Tex Stack Exchange. URL: `http://tex.stackexchange.com/a/40531` (version: 2015-04-09).

[4] Brian Kell (http://tex.stackexchange.com/users/15552/brian-kell), *Is there a "contradiction" symbol in some font, somewhere?*, Tex Stack Exchange. URL: `http://tex.stackexchange.com/a/59617` (version: 2012-06-12).

[5] domwass (http://tex.stackexchange.com/users/1376/domwass), *What's the quickest way to write "2nd" "3rd" etc in LaTeX?*, Tex Stack Exchange. URL: `http://tex.stackexchange.com/a/4119` (version: 2011-04-05).

[6] Heiko Oberdiek (http://tex.stackexchange.com/users/16967/heiko-oberdiek), *Colored symbols*, Tex Stack Exchange. URL: `http://tex.stackexchange.com/a/85035` (version: 2012-11-30).