Formatting Tips and Tricks

Some potentially helpful examples

John W. Lockhart
Red Hat, Inc.
`lockhart@{oco.net,redhat.com}`

Optional Second Author
Second Institution
`another@address.for.email.com`

Abstract

This example paper contains tips and tricks to ensure that what you write is what appears in the *Proceedings* with as little editing as possible. The most important parts are at the end; please read them. (Okay, okay: Section 8 and Figure 1.)

If you are new to LaTeX, please read this paper in its entirety, and check out its source and any other `.tex` files in the `EXAMPLE` directory.

If you have a paper from the Linux Symposium or GCC Summit (2002–2004), and would like to crib from its final formatting, please drop me a note and I'll be happy to send along the edited source. Likewise, if you would like a copy of the final edited form of this year's source, just let me know.

The tree was created based on the information on the conference website. If you don't have a subdirectory, create one along the same lines. Blank materials are in the `TEMPLATES` directory; `ProtoMake` and `Blank.tex` are probably the most interesting files. Likewise, if your Abstract was available when I looked, it has been included. Feel free to edit it; it's just there to get you started and to provide an example of how to properly include files should you need to.

Many thanks go to Zack Weinberg for studying prior years' templates and proceeding to write the `ols.cls` class and other crucial bits of infrastructure. The new system should provide for a lot more flexibility than the old.

1 Simple Formatting Tricks

LaTeX is just a fancy markup language. . . *most* of the time.

Some of the more common font and layout conventions follow:

- `texttt` produces `typewriter` style.
- `textit` produces *italics*.
- `textbf` produces **boldface**.
- `textsc` produces SMALL CAPS.
- *`Font`* **STYLES** can be ***combined***[1]

Paragraphs can be awfully messy in the source, and even have comments interspersed. Be careful with percent signs—75% of the time you'll accidentally comment out the rest of the text on the line.

[1]Often eye-breakingly. Restraint is Good.

Unescaped dollar signs will put you into math mode, so be likewise careful. Of course, that's sometimes exactly where you *want* to be.

Tildes do not produce tildes in LaTeX—think instead of HTML's `&nbsp;` and you'll get the picture. Instead, you can use `\~{}` or `\textasciitilde` to produce a tilde. Table 1 provides a list of characters that require special handling. Note that tables may "float"—that is, LaTeX might move your table to a place where it all fits on a single page, rather than putting it exactly where you have included it in your source. Be aware that it's easier to include references to tables and figures than it is to force each into a particular position and adjust the surrounding typesetting.

Char	Command	Otherwise
#	`\#`	argument number
$	`\$`	toggle math mode
%	`\%`	comment: ignore rest of line
&	`\&`	tabstop
_	`\_`	subscript in math mode
{	`\{`	open environment
}	`\}`	close environment
˜	`\~{}`	non-breaking space
~	`\textasciitilde`	non-breaking space
\	`\textbackslash`	begin command

Table 1: LaTeX characters that require special handling

1.1 New Macros

A number of macros based on the `url` package are available for this year. They are:

- `ident` – intended for identifiers, `\ident{some_text}` sets the text in `tt` and may break the line at any punctuation. Spaces are deleted.
- `lident` – intended for long identifiers, this works the same as `ident`, but sets the text in a smaller font.
- `code` – intended for short excerpts of code, this works like `ident`, except that spaces are preserved. Lines are not broken on spaces.
- `lcode` – intended for longer excerpts of code, this works like `code`, except that text is set in a smaller font. This probably does not work correctly for multi-line code fragments; consider using the `cprog` package for that.
- `brcode` – intended for excerpts of source code, this works like `code`, except that line breaks may occur at spaces.
- `lbrcode` – intended for excerpts of source code, this works like `brcode`, except that text is set in a smaller font.

Examples are shown in Table 2.

2 Typesetting conventions

You shouldn't have to worry too much here, but I'll illustrate a few things.

Quotation marks, both 'single' and "double," look good in body text, while other `"styles"`

- `\ident{a_long_identifier}` — this example in turn yields `a_long_identifier`
- `\lident|an_even_lon ger_identifier|` — this in turn yields `an_even_longer_identifier`
- `\lcode{int un_useful(int *a) { return *a; }}` — this yields `int un_useful(int *a) { return *a; }`
- `\lbrcode{int un_useful(int *a) { return *a; }}` — this yields `int un_useful(int *a) { return *a; }`

Table 2: Examples of New Macros

might look better for other uses. Note that when you're typesetting for a compiler, punctuation goes outside the `"quotation marks",` but punctuation is placed *inside* the quotation marks for "narrative."

There are multiple flavors of dashes—the em dash, the en–dash, the oft-used hyphen, and the minus sign (math mode: $2x-3$). Note that the preceding sentence contains them all.

2.1 Choices for uniformity

For source code, we have chosen the common style of not beginning a line with a comma. The compiler doesn't care, but keeping the printed page consistent between papers is useful.

Identifiers may need to be split between lines, so we use a typewriter font and mark up the string appropriately: `sys_sched_yield()` or `A_REALLY_LONG_IDENTIFIER_THAT_NEEDS_TO_BE_THIS_LONG` would be good examples[2]. To tell LaTeX that an unhyphenated line break is okay if required, just use `\linebreak[0]`.

2.2 Points of English

A few nitpicks:

1. *it's* is a macro which expands to *it is*. It has no other meaning.
2. *its* is possessive.
3. Items in a series are: *a*, *b*, and *c*. Never *a*, *b* and *c*. This rule makes it much simpler when you must use complex values of (for example) *b*. For truly long constructs, you may use a semicolon as a delimiter rather than a comma.
4. Some phrases should be hyphenated—for instance, when you're using an adjective to modify another adjective, or a noun that appears before another. A high-performance system; a win-win situation; a high-level loop transformation; a slow-moving train, but a slowly moving car; that sort of thing. Most of the time, people will still be able to parse the results easily if the sentence isn't perfect.
5. Be happy, know your homonyms. There, they're, their. To, two, too. Your, you're. And so forth. Spelling checkers show their limitations on this…

Of course, proofreading is a wonderful thing, and every bit of it you (or any guinea pigs you can persuade) do is a Good Thing. I'll correct what I notice, but I have only two eyes and there's a lot of margin-crunching formatting to

[2] Alternatively, see the macros in Section 1.1.

be done. There are certain times, often with non-native speakers, where I'm not clear on the meaning. If I catch something like that in time, I'll ask; if not, chances are that I'll keep my hands off of the section in question so as not to insert a woefully incorrect meaning.

3 Tools

It helps to have the following installed on your system:

- `tetex`. The most common TEX package for Linux.
- `dviutils`. Required for building the 2005 Proceedings. Can combine DVI files as well as other useful tasks.
- `transfig`. Graphics in `.fig` format, useful for figures.
- `dia`. Also useful for figures.
- `ImageMagick`. Great for photographs and graphics manipulation & conversion.
- `xpdf` or `acroread` for viewing PDF files. Other viewers can also do a nice job.
- Utilites often found in `tetex`, but which your distribution may have packaged separately: `xdvi`, `dvips`, `pdflatex`.
- `ghostscript` for handling Postscript.

4 Examples

Some examples from previous conferences have been included in this package; hopefully they'll be useful in handling code examples. Reducing everything to `footnotesize` or setting it `verbatim` won't magically make it fit on the page, alas. Have a look in the `EXAMPLE` directory to find these items:

- `bibliography.tex`, `bibliography2.tex`, and `references.tex`. Different ways of citing any relevant works external to your paper.
- `conditional.tex`. If you have LATEX code that works only by itself and need to do conditional processing, here's an example.
- `complexCode/complexFigure.tex`. An example of a complex figure containing side-by-side C code.
- `figures.tex`. Different ways of doing figures.
- `includegraphics.tex`. Different ways to include graphics.
- `legalese.tex`. Legal disclaimers.
- `multipleAuthors.tex`. Formatting examples for multiple authors.
- `tables.tex`. Different ways to do tables.

4.1 Bad Examples

A prior year's paper gave the example of setting `verbatim` sections in `tt`. Repetitiously and redundantly enough, that's the default. So, please, no instances of

```
{\tt
\begin{verbatim}
 ...
```

Corrected. You might, however, wish to do something like this instead:

```
\begin{small}
\centering
\textbf{Corrected.}  You ...
\begin{verbatim}
  ...
```

Of course, check the source of this document (`EXAMPLE/myPaper.tex`) for more ideas. Valid font sizes, for instance, include `normalsize`, `small`, `footnotesize`, `scriptsize`, and `tiny`. Please don't use anything larger than `normalsize`.

Another extant bad example is the practice of ending paragraphs with a double backslash (`\\`) *and* a blank line. This creates unwanted, superfluous whitespace between paragraphs. LaTeX is, believe it or not, supposed to be easy. Just leave one or more blank lines between paragraphs and you'll be fine.

5 Style packages

For 2005, we are no longer using the `combine` package. You will find some additional useful packages in the `Texmf` directory, however. The empty papers are set up to use the `url`, `zrl`, and `graphicx` packages by default, in hopes that this will be useful for most papers.

You may also find it helpful to set the `TEXINPUTS` environment variable as follows:

```
export TEXINPUTS='.//:${LOCALTEX}//:'
```

Adding the above to your `~/.bashrc` can save you the trouble of typing it for future runs.

To build your paper, you should be able to `cd` to the toplevel directory (the one that contains your individual directory) and type the following at a shell prompt:

```
DIRS=yourname make
```

Ambitious authors are encouraged to install the `dviutils` and `pdftk` packages and type `make` from the top-level directory. If all goes well, you'll get something that looks quite like the finished *Proceedings*.

6 Graphics and Symbols

For importing graphics, don't forget to omit any file extensions. That's because `latex` and `pdflatex` look for different formats. The output formats we generate are PDF, PS, and DVI; you will thus want to generate both EPS and PDF copies of any figures that use structured graphics.

The easiest ways to get special symbols such as Registered® and Trademark™ is to use the LaTeX2e `\text` constructs: thus, `\textregistered` and `\texttrademark`.

7 TeX References

If you aren't familiar with LaTeX, there are many sources of information available. Your distribution might have additional documentation in `/usr/share/texmf`, or you might find manuals for a package (such as `cprog`) out at `http://www.ctan.org`.

If you are completely new to TeX and LaTeX, you will probably find it highly useful to visit `http://www.tug.org/` and especially `http://www.tug.org/begin.html` for online and paper references.

For a free and extremely useful document, try: `http://www.tug.org/tex-archive`

`/info/lshort/english/lshort.pdf`. Note that translations[3] are available, for those more comfortable in something other than English: `http://www.tug.org/tex-archive/info/lshort/`

I tend to use *A Guide to LaTeX* (Kopka & Daly, ISBN 0-201-39825-7) and the *LaTeX Graphics Companion* (Goossens, Rahtz, & Mittelbach) the most these days.

You are also welcome to send questions to me at `lockhart@redhat.com` (work) or `lockhart@oco.net` (home).

As usual, please refrain from submitting anything remotely resembling a Microsoft Word `.doc` file…`<grimace>`. It's a *lot* easier for me to fix up plain ASCII text and convert/insert accompanying graphics, if you find yourself terminally confused or in a dire emergency.

SUBMITTING A PAPER

```
cd OLS2005
make clean
tar zcf yourLastName.tar.gz \
    yourLastName
```

E-mail the resulting tarball to `papers@linuxsymposium.org`.

Figure 1: Submitting a paper

8 Simple rules to keep your formatting team happy

1. To submit your paper, just `make clean` in your directory, `tar` it up, and send the resulting gzipped tarball to `papers@linuxsymposium.org` or `papers@gccsummit.org`, as appropriate. See Figure 1 for an example.

2. Updates. If you need to change something, please send both a patch and an updated tarball. The most convenient form depends on how many changes have been made since you submitted your paper. However, if your change is trivial—a line or two, for instance—a simple email will do.

3. Use the existing directory structure, please. The directory names are intended to be the last name of the presenter (lowercase, punctuation omitted); the main paper should be `lastname.tex` and any additional files should be `lastname-file.extension`. This is basically to keep the file owners straight, and to allow us the option to instruct LaTeX to search the entire (sub)directory hierarchy for input files. You don't want someone else's file by mistake, right? Putting your name on it helps to keep things straight. The same goes for `\label{}` and `\ref{}` commands.

4. Omit file extensions and pathnames in your LaTeX source, please. By omitting the path and just saying `\input{lockhart-abstract}`, a paper can be built from both its directory and from its parent directory. For graphics, omitting the extension lets `latex` or `pdflatex` pick its preferred input format for the best possible results.

5. No proprietary document/graphics formats, please. This especially means MS Office, Visio, or other such tools. LaTeX can, however, import EPS and PDF, if you can save in those formats.

[3]French, for instance: `http://www.tug.org/tex-archive/info/lshort/french/flshort-3.20.pdf`; note also that this section of the Example paper shows different ways of handling URLs.

6. Originals, please. For example, if you have photographs, send along the full-resolution JPG (crop out any undesired elements if necessary, but use the maximum resolution). For diagrams, please send the XFig or Dia files. This ensures the best possible print quality. Printing will be in black and white, but the online PDF's will be in full color. Your screen is probably about 72dpi, but the typesetter is probably using something that's at least 1200dpi. The more resolution, the better. (If, however, your originals are outrageously huge, feel free to ask!) Since hardcopy will be printed in Ottawa, the papersize will be North American "letter." Please keep that in mind if you are concerned about page breaks and such.

7. Do ***not*** use sans-serif fonts, or go changing global font sizes. We're using 12-point Times Roman for body text. Likewise, please don't go haywire with italics. I once received a huge collection of tables, each of which set the font size and face on an item-by-item basis. *Incorrectly*.

8. Those of you who like to begin lines of code with commas: as previously mentioned, we're typesetting the code with the comma attached to the preceding identifier (as most publishers do). Feel free to post your preferred version to the web and to refer to it in the paper.

9. If possible, please avoid trivial new macros. Should you need to add something, though, please use `\providecommand` rather than `\newcommand`, and try for a relatively unique name (papers tend to blur together during long editing sessions).

10. Trivia note: generally speaking, it takes longer to edit a submission from a TEXspert than plain, unmarked ASCII. If you consider yourself a LATEX expert and love to write fancy new commands, please consider contributing clean-ups or well-tested new features for the infrastructure rather than customizing the daylights out of your submission. Thanks!

This paper builds correctly using the tetex-2.0.2-14FC2.2 package on Fedora Core 2, and the Fedora Core 3 tetex package. Please note that if you are using FC3, you may wish to update your `urw-fonts` package to 2.2-8 or better before viewing PDF files.

Other distributions haven't been tested, but should work. If you run into problems, please let me know.

And remember, it's only typesetting, not rocket science. Or hacking compilers or kernels. `:-)` Have some fun along the way...