

Formatting Tips and Tricks

Some potentially helpful examples

John W. Lockhart
Red Hat, Inc.

lockhart@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 10 and Figure 4.)

If you are new to L^AT_EX, please read this paper in its entirety, and check out its source and any other .tex files in the EXAMPLE directory. Reading EXAMPLE/EXAMPLE.tex side by side with this PDF should be helpful.

If you have a paper from the Linux Symposium or GCC Summit (2002–2007), 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. The 2007 papers and sources should be publicly and anonymously available again at some point—see <http://people.redhat.com/lockhart/ols> for updates. The final, fully edited form of this year's source will appear at or around the time of the conference.

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.

Please note that we generate both PDF and Postscript output for the papers, so both pdf_latex and latex will be used in the process. This happens automatically if you use the supplied build system; you just need to ensure that both Postscript and PDF versions of your graphics are available. This is discussed in Section 7.

Note that using a non-Linux system is possible, but will likely cause you stress, misery, and considerable extra effort to solve all the possible compatibility issues. This is especially important when it comes to graphics and embedded fonts.

Bitmapped graphics are forbidden. Unless you have a photograph or a screenshot, no bitmap graphics, please. This includes bitmaps which have been converted to EPS. Start with structured graphics: Inkscape,¹ Dia, XFig, gnuplot,² and OpenOffice.org³ are all viable choices for creating diagrams. Programs such as the GIMP or Powerpoint are not. See Section 7 for one example of why we forbid bitmaps.

1 New This Year

- Authors can submit papers and updates directly into the formatting source control system. See Figure 1 for basic directions, and <http://people.redhat.com/lockhart/ols> for any updates.

You will need an account on FedoraProject.org since the current year's papers are not publicly available until after the conference. Please visit <https://admin.fedoraproject.org/accounts/user/new> to create an account if you do not have one. Note that the FedoraProject site hosts many projects, and we do not run the site. Thus we cannot create an account for you; you must do that part yourself. This especially includes the GPG and SSH keys.

Then make a request to join the **ols** project via the web interface (see “Apply For A New Group”). If

¹Saving as PDF seems to work better with Inkscape.

²set terminal postscript eps color is the command you want to use with gnuplot.

³As of 2007, OO.org required that you turn off page numbers, save as a PDF, run pdfcrop on the saved output, and then run pdftops to get EPS output.

this doesn't work, try sending mail with your FedoraProject username to `ols-sponsors` over at `fedoraproject.org`, to the `papers2008` address, or to me (in that order of preference, since multiple people can approve the request).

Once you have been added to the project, please see the SVN checkout directions at <http://people.redhat.com/lockhart/ols/SVN-tips.html> for details on the basics of using SVN with GCC papers. Figure 1 shows the basics.

- The current reference system for GCC and OLS builds is running Fedora 8. The `dviutils` package is required for full builds, and may be found in SVN if needed.

2 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*⁴

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.

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 —they instead produce non-breaking spaces, akin to HTML's ` ` character. Instead, you can use `\~{}` or `\textasciitilde` to produce a tilde. Table 1 provides a list of characters that require special handling.

⁴Often eye-breakingly. Restraint is Good.

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
#	<code>\#</code>	argument number
\$	<code>\\$</code>	toggle math mode
%	<code>\%</code>	comment: ignore rest of line
&	<code>\&</code>	tabstop
_	<code>_</code>	subscript in math mode
{	<code>\{</code>	open environment
}	<code>\}</code>	close environment
~	<code>\~{}</code>	non-breaking space
~	<code>\textasciitilde</code>	non-breaking space
\	<code>\textbackslash</code>	begin command

Table 1: \LaTeX characters that require special handling

2.1 Customized Macros

A number of useful macros based on the `url` package are available. Although they do not work in some floating environments, they should be used everywhere else—they're easier to type and produce better results than the alternatives. They are:

- `ident` – intended for identifiers, `\ident{some_text}` sets the text in `typewriter` 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.

If you have created an account, sent mail to lockhart over at redhat dot com for access, and received a welcome note, proceed as follows. This example assumes that your FedoraProject username is itsme, and avoids checking out all the papers in the tree.

Note that if your local username differs from your FedoraProject username, use FedoraUser@fedorapeople.org instead in the SVN checkout command below. From time to time, it may be useful to “make update-templates” to ensure that you are using the latest and greatest framework.

```
# First check out the top-level directory
$ svn checkout svn+ssh://fedorapeople.org//home/fedora/lockhart/ols/gcc2008 gcc2008
--username itsme --non-recursive
# Alternatively: svn checkout svn+ssh://itsme@fedorapeople.org/ ...
$ cd gcc2008
# Use the makefile to grab the rest of the templates
$ make update-templates
# Discover which directory has been set up for your paper.
# The tool is especially useful if you have more than one paper,
# or there is another author with the same last name.
$ ./Showtree.rb | grep -i MyLastName
# Then update to check out the skeletal paper.
$ svn update mylastname
$ cd mylastname
# Start editing. To submit or update, just ensure that you're
# in the gcc2008/mylastname directory, and then:
$ svn update
$ svn ci -m 'a short summary of what you changed'
```

Figure 1: Setting up and Using Subversion with GCC and OLS Papers

- `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.

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 for 2005. We continue to use this system, with a few minor tweaks to its features. So if you have already started writing and

are using last year’s templates, you should be fine with this set.

The document class for your paper can be one of `final`, `galley`, or `proof`. The default value is `final`. Some folks might want to use the `galley` option during the writing process; others might find `galley` to be confusing. In any case, please be sure to submit your paper using the `final` class. The differences are:

- `galley` — All “this doesn’t fit” warnings are suppressed, and references are printed as textual keys rather than as numbers. This is because when you’re working on the paper and have the TeX in front of you along with a printout, it’s often easier to use “See Section [whatever-I-called-it]” than “See Section 8.” That way, you can just search the `.tex` file for the section by name to make changes, since there’s no “Section 8” in said `.tex` file...

- `\ident{a_long_identifier}` — this example in turn yields `a_long_identifier`
- `\lident|An_Even Longer Identifier|` — this in turn yields `An_EvenLongerIdentifier`
- `\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

- `proof` — All “this doesn’t fit” warnings are active, as are references. Overfull hboxes make ugly black blobs in the margin. You can use this mode to tidy up formatting after you’re done writing. This mode is the same as the article class’s “draft” mode.
- `final` — All warnings and references are active, and the paper produced will be similar to the one which will be published (but without headers, final page numbers, and such).

3 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” 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.” If you are one of those who like to introduce a bunch of “terms,” each one in “quotes,” please use italics for the first instance of each *term* instead.

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.

3.1 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.⁵ To tell \LaTeX that an unhyphenated line break is okay if required, just use `\linebreak[0]`. Whenever possible, we strongly encourage use of the macros in Section 2.1 instead—they’re both easier to use and easier to read.

3.2 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. This is especially useful if item *b* itself contains commas.
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 even if the sentence isn’t perfect.

⁵Alternatively, see the macros in Section 2.1.

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 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.

4 Tools

It helps to have the following installed on your system:

- **tetex.** The most common T_EX package for Linux. Related useful packages include **dviutils**, **xdvi**, **dvips**, and **ghostscript**. Be aware that certain distributions package `pdflatex` separately.
- **transfig.** Graphics in `.fig` format, useful for figures.
- **dia.** Useful for figures. Other commonly used graphics programs include **transfig**, **xfig**, **inkscape**, and **OpenOffice.org transfig**. Please note that your export options are crucial, and that we recommend sending along the original, native file as well. You should aim for portability: for instance, certain OpenOffice versions let you choose between EPS with Pango fonts, or EPS — you want the non-Pango EPS for portable printing. As of 2007, the best output from OpenOffice is generated by turning off page numbers, saving as PDF, and running `pdfcrop` on the saved PDF. `pdftops` can then generate EPS from the PDF.
- **ImageMagick.** Great for photographs and graphics manipulation & conversion, especially the `convert` program.
- **xpdf**, **evince**, **kpdf**, or **acroread** for viewing PDF files. Other viewers can also do a nice job.

- **pdftk** if you wish to concatenate PDF pages or perform several other tasks with PDFs. This package is required for building the final Proceedings.
- **gnuplot** for drawing graphs based on large quantities of numeric data.
- **dviconcat** is required for building the entire Proceedings, and can be had from the *dviutils* package, available in SVN, or available as source, Debian, or SuSE packages (possibly other distros).
- **emacs** is my editor of choice, in part due to its “LaTeX Fill” mode, which does a very nice job of handling markup and syntax, with automatic line wraps. Please check to see if your editor of choice has a mode for editing L^AT_EX files; it could save you from tracking down syntax errors and such. (I also use and enjoy `gvim`, especially its *diff* mode.)

Beware any program that cannot export structured graphics, or has troubles respecting standards. These are most commonly found on non-Linux systems; with some of these, you may be better off printing to a hypothetical Apple Laserwriter on `FILE:` and converting the proceeds than in trying to rely on the program doing anything remotely sane and portable.

5 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 L^AT_EX 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. Please note that people want to read technical information, not disclaimers; thus, we set any section of disclaimers in small fonts. To make the Demands Of LawyersTM easier to deal with, we provide `\LegalTM` and `\LegalR`. The latter looks like This[®].
- `multipleAuthors.tex`. Formatting examples for multiple authors.
- `tables.tex`. Different ways to do tables.

5.1 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.

6 Style packages

You will find some additional useful packages in the `Texmf` directory. 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. The build system uses this setup by default.

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*.

7 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 macros provided in `ols.cls`: `\LegalR` and `\LegalTM`. You may also use the \LaTeX 2 ϵ `\text` constructs: thus, `\textregistered` and `\texttrademark`. If you need to use asterisks or the Registered or Trademark symbols in a title—please avoid doing so, it's ugly. If you can't avoid it, the macros `\titleLegalTM`, `\titleLegalR`, and `\titleStar` are the way to insert the marks and avoid any havoc in the Table of Contents.

```
% By omitting the extension,
% - pdflatex finds jwl-page-fig.pdf and jwl-col-fig.pdf
% - latex finds jwl-page-fig.eps and jwl-col-fig.eps

% Full pagewidth figure, spanning both columns:
\begin{figure*}
\includegraphics[width=0.9\textwidth]{jwl-page-fig}
\caption{The caption appears beneath the figure}
\label{jwl-page-fig-label}
\end{figure*}

% Single-column figure:
\begin{figure}
\includegraphics[width=0.9\columnwidth]{jwl-col-fig}
\caption{Captions appear beneath the figure}
\label{jwl-col-fig-label}
\end{figure}

% Example reference:
See Figure~\ref{jwl-col-fig-label} for single-column use.
Remember that the tilde is a non-breaking space.
```

Figure 2: How to use `includegraphics`

We generally try to leave a small margin around each figure, and thus use constructs such as those in Figure 2 for a full-width or full-column picture.

Please note that the `.eps` and `.pdf` extensions have been omitted. `latex` will automatically search for the former; `pdflatex`, the latter. Adding an extension will break one tool or the other. For the full page width, use `figure*`; otherwise, use `figure`. See Figure 2 for the example.

This year's Proceedings will be in color (or, if you prefer, *colour*). If you choose to use color graphics and figures, please do so effectively, drawing attention to important information and distinctions in your paper. A single box or letter doesn't usually add much information, but does squander resources needlessly.

Please avoid using bitmapped graphics unless you have no other choice (for instance, a JPG photograph). Structured graphics always produce better printed output. Figure 3 provides an example.

8 T_EX References

If you aren't familiar with L^AT_EX, 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`) at <http://www.ctan.org>.

If you are completely new to T_EX and L^AT_EX, 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 translations⁶ 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 L^AT_EX* (Kopka & Daly, ISBN 0-201-39825-7) and the *L^AT_EX Graphics Companion* (Goossens, Rahtz, & Mittelbach) the most these days.

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

⁶French, 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.

NO BITMAP GRAPHICS



Figure 3: Above are a PNG and an EPS. Easy to tell apart when magnified without smoothing features.

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 gcc2008
make clean
tar zcf yourLastName.tar.gz \
    yourLastName
```

E-mail the resulting tarball to `papers2008@gccsummit.org`. Please remember the `make clean`; to mail anything that is built anyway is a waste of bandwidth.

Figure 4: Submitting a paper

Modify `GCC2008/author/Makefile.inc` in the following fashion (keeping in mind that those aren't leading spaces, but tabs):

```
## Add any additional .tex or .eps files below:
auth/auth.dvi author/auth-proc.dvi: \
    auth/auth.tex author/auth-abstract.tex \
    auth/auth-figure1.eps \
    auth/auth-figure1.pdf \
    auth/auth-figure2.eps \
    auth/auth-figure2.pdf

auth/auth-%.pdf: auth/auth-%.eps
    epstopdf $<
```

Doing a `make` on the toplevel will get you an `author-proc.pdf` file, which is pretty much how it'll look in the finished Proceedings. Feel free to delete the other authors' directories in your tree.

9 Build Issues

9.1 PDF from EPS

To make PDF graphics from your EPS files, you can adjust your `Makefile.inc` to run `epstopdf` (which is generally part of the `tetex` package on most distros) automatically. Here's an approach that would work for an author named `auth...`

10 Simple rules to keep your formatting team happy

1. To submit your paper, either use SVN (which is the preferred method), or just run `make clean` in your directory, `tar` it up, and send the resulting gzipped tarball to `papers2008@gccsummit.org`. See Figure 4 for an example.
2. Updates. If you need to change something, please make your changes in SVN. If for some reason that

is not possible, 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. Although you can convert a bitmap into EPS/PDF, please don't do so. If you have screenshots or photos, PNG or JPG can be used. Everything else should be structured graphics only.
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 a mixture of color plus black and white, but the online PDFs will be in full color. Your screen is probably about 72dpi, but the printing press 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 Times Roman for body text, in a consistent size that's comfortable for hardcopy. 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 \TeX spert 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 on Fedora 8 and Red Hat Enterprise Linux 5. Other distributions haven't been tested, but should work. If you run into problems, please let me know so that I can try to make it work.

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