Hints & Tricks

Whatever is Wrong with my LaTeX File?

Sebastian Rahtz

1 Introduction

Contrary (perhaps) to what many TeX people experience, much of the LaTeX that I have to untangle is not written by me. At Elsevier Science we accept LaTeX files for more or less any of our 1200+ journals, and our production editors have to coerce the submissions into a standard form so that we can apply our journal-specific styles. Along the way, many problems can arise, and we hold in-house training sessions to discuss techniques for finding bugs in other people's LaTeX. These notes arise from those sessions, and are offered as a light-hearted reminder to LaTeX writers and editors alike about some of the ways the agony of using our idiosyncratic system can be lessened. Following a felicitous parallel drawn by the *TUGboat* reviewer of this article, think of this like those posters on the doctor's wall which you read while waiting to see the specialist. It is not a serious guide to TeX debugging, for which I am not qualified, and which would require a very large book indeed...

Perhaps these musings[1] will stimulate others to write about *their* working methods in *TUGboat*.

2 Golden rules

If you do not take the following precautions, you might as well give up writing or editing LaTeX now:

1. *Look* at TeX errors; those messages flashing across the screen are not some kind of screen saver.
2. Be prepared to read the log file too; did you realize it has extra information? Specifically, it will list characters missing from a font.
3. OK, so you ignored those two rules; but at least realize you *have* a log file, and take it with you when you visit the doctor.
4. Lay out the source sensibly; how can you find errors if your input is one long line of mixed macros and text?
5. Use syntax checkers; there are many of these: I use *lacheck*, from the authors of Emacs AUCTeX, and the one built into Eddi4TeX, but there are others. For LaTeX especially, it is a god send to have the missing `\end{enumerate}` spotted for you.
6. LaTeX has several packages to help show you what it is working with: `showkeys` shows you the labels you define; `syntonly` will run a LaTeX file fast, ignoring fancy typesetting; the `listfiles` command lists the macro files that were used at the end (handy for checking versions), and the `draft` option will show overfull boxes and all manner of other things for some packages.
7. If you are a confident macro programmer, be aware of the many TeX primitives that can help you: set `\errorcontextlines` to give more context for help messages, use `\message` to put in diagnostic messages, try `\meaning` to find out what a macro really *is* defined as, rather that what you assumed it was. Don't despair at the amount of verbiage `\tracingall` gives you — there is gold there if you dig deep enough.
8. Remember primitive programmer's debugging techniques; if all else fails in your quest to see why LaTeX dies with that weird error in your 10000 line file, move `\end{document}` gradually back up the file from the end until it *does* work, and then stare at the 10 lines which you know provoke the error, with a wet towel around your

[1] An earlier version was published as part of the editorial in *Baskerville* 5(5), and is used with permission of the UK TeX Users Group.

head. It is faster than reading all 10000 lines over and over again hopelessly...

9. *Do not* mail the LaTeX development team, or other package authors, every time TeX gives you an error prompt; you'll irritate hard-pressed volunteers working in their spare time. If you wait until you have a *good*, well-documented, repeatable, error condition that your friends get too, *then* you can report it, and likely get a friendly reply and a fix.
10. Read before you Write. There are many *excellent* books about TeX and LaTeX that you can buy and read, as well as the freely available 'Frequently Asked Questions' document (make sure you get the UKTUG version, as it is considerably changed and enhanced from the original). You *cannot* use LaTeX without a manual.

3 Examples

3.1 Layout

Did you think I was joking about laying out your text in a readable fashion? Can you easily find the error in this example?

```
\begin
{document}\baselineskip=12pt\newcommand
{\F}{Fig.~}\newcommand {\w}{\omega
}\newcommand {\k}{\xi }\newcommand
{\p}{\phi
}\maketitle\thispagestyle{empty}\centerline
{\bf \underline{Abstract}}\vskip
6ptA probabilisticoptimal design
methodology for complex structures
using the existing probabilistic
optimization techniques.  \vskip
12pt\centerline{\bf
\underline{Nomenclature}\vskip 6pt
\begin{tabbing}\( A
\)\hspace{0.45in} \=:
Transformation matrix\\\( a_i \)
\>: Gradient of performance
function with respect \\$\hskip
1.25in$ to $i^{th}$ random variable
\\\( b \) \>: Design variable
vector\\\( {\it CDF} \) \>:
Cumulative distribution
function\\\( {\it COV} \) \>:
Coefficient of variation \\\( C_x
\) \>: Covariance
```

That is, of course, an artifical example, but one does come across files which look a bit like this. Common sense (and the LaTeX manual) will suggest that replacing code like:

```
\vskip 3pt\noindent{\bf \underline{Safety
Index Interpolation}}\vskip 1pt
```

with

```
\section{Safety Index Interpolation}
```

will considerably aid readability and maintenance. It is a curious fact that some files sent in to Elsevier journals purporting to be LaTeX are little more than plain TeX with `\documentstyle` inserted at the front, and the above is not unusual. It also arises when a frustrated LaTeX user cannot work out how to make the `\section` command do what is required, so brute force is used at the last moment.

Do not stop at simply choosing rational places for line endings; is this

```
\title{Some dull results}
\author{My AlterEgo}
...
and that was the last paragraph.
\section{Another section}
\begin{enumerate}
\item \emph{Look} at \TeX\ errors;
those messages flashing across
the screen are not some kind of
screen saver.
\item Read the  log file too; did
you realize it has extra
information? Specifically, it will
list characters missing
from a font.
\end{enumerate}
```

as easy to read as this?

```
%-------------------------------------
\title
            {Some dull results}

\author
            {My AlterEgo}
%-------------------------------------

....
and that was the last paragraph.

%-----------------------------------------
\section{Another section}

\begin{enumerate}
 \item \emph{Look} at \TeX\ errors; those
       messages flashing across the screen
       are not some kind of screen saver.
 \item Read the  log file too; did you
       realize  it has extra information?
       Specifically, it will list characters
       missing from a font.
\end{enumerate}
```

Again, this seems trivial, but consistent and readable layout of the code is well worth the trouble; some intelligent editors (like Gnu Emacs in AUCTeX mode) can do almost all of it automatically.

3.2 Syntax errors

TEX error messages are not as obscure as we sometimes think; here is an example where the puzzling output is all explained in the log file:

```
{This is not so bad,
\bfseries\ttfamily hello?}
{This is not so bad, \scshape
Hello \bfseries Goodbye?}
{\it\bf\Large byebye}
\end{document}
```

Why do we not see bold typewriter or bold small caps? Because the fonts do not exist, and LATEX tells us it has had to make substitutions as best it can:

```
LaTeX Font Warning: Font shape
 `OT1/cmtt/bx/n' in size <10>
  not available
(Font)        Font shape `OT1/cmtt/m/n'
 tried instead on input line 4.

LaTeX Font Warning: Font shape
 `OT1/cmr/bx/sc '   undefined
(Font)        using `OT1/cmr/bx/n'
 instead on input line 6.
```

What more could you ask? Regular LATEX users must learn to understand these New Font Selection Scheme messages, as they are a crucial part of LATEX 2_ε.

Now let us look at a bad file which is quite easy to understand:

```
\documentclass{article}
something
\begin{document}
hello \( a=
\end{documen
```

LATEX says of this, in an unusually clear way:

```
! Missing $ inserted.
<inserted text>
                $
l.4

?
)
Runaway argument?
{documen
! File ended while scanning use of \end.
<inserted text>
                \par
<*> bad

?
```

though the 'missing $' is a bit confusing when what it meant was 'missing \)'. *lacheck* does a much better job:

```
"bad.tex", line 5:
      <- unmatched "\end{documen}"
"bad.tex", line 3:
      -> unmatched "math begin \("
"bad.tex", line 5:
      <- unmatched "end of file bad.tex"
"bad.tex", line 2:
      -> unmatched "\begin{document}"
```

However, it sees nothing wrong with this:

```
\documentclass{article}
\begin{document}
Funnies: \dag, \AA and \"
\section{Introduction}
\end{document}
```

about which LATEX says:

```
! Argument of \@xdblarg has an extra }.
<inserted text>
                \par
<to be read again>
                   }
l.5 \section
            {Introduction}
?
```

How long did it take you to spot the problem? Can someone suggest a technique other than towel-round-the-head staring to catch it?

3.3 Hyphenation

If hyphenation is your bugbear, do you understand the difference between the following large heavy animals?

```
rhinoceroses
\showhyphens{rhinoceroses}
\hyphenation{rh-ino-cer-os-es}
rhinoceroses
\begin{sloppypar}
rhinoceroses
\end{sloppypar}
rh\"inoceroses
\fontencoding{T1}\selectfont
rh\"inoceroses
\par\hskip\z@skip
rhinoceroses
```

Remember that:

1. TEX may need help hyphenating the word; give it clues;
2. If you want justification at all costs, set the right parameters — `sloppypar` goes too far, using *very* lax settings, but it works;
3. If you put accents in words, hyphenation dies ...
4. ... unless you use T1 encoding, which cleverly transforms `\"i` to an 8-bit character internally

so that TEX proceeds happily (but remember that you need 8-bit hyphenation patterns to do a proper job);

5. The first word of a paragraph will not hyphenate. Insert something harmless to bypass this law.

3.4 Frequently encountered pitfalls

I expect all my readers have written something like this at some time:

```
\begin{figure}
\label{fig1}
\caption{This is a caption}
\end{figure}
```

and wondered why the labels are wrong. It is *not* the figure environment which sets labels, but the `\caption` command; what the example above will do is set the label 'fig1' to the value of the most recent section, equation, list item or whatever.

Do the new LATEX2e packages puzzle you? Why doesn't this work:

```
\usepackage{graphicx}
\begin{document}
This is \rotatebox{75}{hello sunshine}
  at an angle
\end{document}
```

Simply because rotation, colour, scaling, and graphics insertion are all device dependent, and LATEX needs to know what dvi driver you have. You probably meant something like:

```
\usepackage[dvips]{graphicx}
```

Lastly, did your TEX just say 'bufsize exceeded'? Maybe the file it was reading came from a Mac? or a word-processor which stored each paragraph as a single long line? If it is a graphic file, it may have come from a Mac package, and TEX is throwing up while searching for a `%%BoundingBox` line. You should realize that DOS, Unix and Mac treat line-endings differently! If you don't have a dedicated utility to fix this, try using *zip* to package up the files, and then *unzip* them, using the flag to convert text files to the local native format.

4 Conclusions

One could go on listing common problems, and mysterious LATEX errors, for many pages. But the fundamental message is that you cannot treat TEX products like the finite and menu-driven offerings from Microsoft. If you write your documents using a computer programmer's assembly language, you are always going to be exploring strange new worlds. If you think you have better ways of spending your time — don't use TEX directly at all. The excellent *Scientific Word* interface to LATEX will spare you most of the pain described in this article, and others are sure to follow.

Choose LATEX with a light heart: If you can keep your head when all about you Are losing theirs and blaming it on you... If you can wait and not be tired by waiting... if you can meet with Triumph and Disaster, And treat those two imposters just the same; ... If you can bear to hear the truth you've spoken Twisted by knaves to make a trap for fools, Or watch the things you gave your life to, broken, And stoop and build 'em up with worn-out tools ... If you can fill the unforgiving minute With sixty seconds' worth of distance run, Yours is TEX and everything thats in it, And — which is more — you'll be a Man, my son!

◇ Sebastian Rahtz
Production Department,
Elsevier Science Ltd,
The Boulevard, Langford Lane,
Kidlington, Oxford OX5 1GB
UK
Email: s.rahtz@elsevier.co.uk