What is a minimal working example?

Christian Faulhammer
Version: 0.6

2009-08-18

Contents

This document was created out of a need to supply newcomers to the newsgroup `de.comp.text.tex` with a collection of examples to demonstrate and clarify the principle of minimal working examples. This text is in the public domain, the newsgroup's version (in German) is currently maintained by Christian Faulhammer and referred to in the regularly posted introduction. Please direct corrections, suggestions and comments to christian(at)faulhammer.org.

Anyone wanting to dive further into the meaning of error messages should have a look on the LaTeX Companion, which is besides not suitable for beginners.

Note concerning the English language version: Initial work and the first complete translation was done by Ulrich Schwarz. Now this has been overtaken by Christian Faulhammer.

1 Acknowledgments

First of all thanks to all members of the newsgroup who repeatedly tell people how important a minimal working example is and hints how to create it. Specifically I want to thank Ulrich Schwarz for the inital translation into English, Hilmar Preuße for the first LaTeX version and Kurt Lidwin for the rework of the latter.

2 What are these minimal working examples (MWEs) you keep talking about?

Creating an MWE is a way of tracking down and fixing bugs, as well as of finding the cause of certain behaviour. MWEs must be as small and simple as possible, yet complete, i.e. they accord to the base structure of a LaTeX document and can be LaTeXed without further additions.

An MWE should be complete and working,

- so that you cannot accidentally omit information important to diagnosing the problem;
- so that the person responding can just copy-and-paste the code to try it out.

An MWE should be small,

- so that responders does not have to concern themselves with long unnecessary lines of code;

- because it will narrow down possible causes of misbehaviour;
- because short examples are well suited for posting;
- because many problems, such as missing brackets, faulty syntax, forgotten switches, etc. will easily be found in the process creating it;
- because short documents will help you not to lose track of what's going on.

Even beginners should have no problems creating MWEs, since it only takes a little patience and common sense. In most cases, you will find the cause of your problems yourself, thereby making posts unnecessary. After this short motivation, we will list the tools and procedures at hand, followed by some examples. We will deal with different sources of problems (wrong package options, packages clashing, faulty syntax, etc.) seperately. In complex cases, all these have to be regarded at the same time.

3 Tools

The basic concept underlying the creation of an MWE is: "divide and conquer". There are several ways of dividing, some of which depend on the structure of the document. First of all, the command `\listfiles` should be added to the document, so that all used files (packages, classes etc.), including version information, will be logged. This information can be important to people replying and should not be omitted.

Also, the usual rule holds: make a backup, and only work on copies, to prevent accidental deleting of material!

3.1 Moving the end

3.1.1 One single file

If the whole source is in a single file, the following technique is useful: suitable advancing of the command `"\end{document}"` will help to coarsely narrow down where the problem is. Doing this, we move upwards by blocks. (This meaning logical blocks, i.e. paragraphs, environments given by `\begin- \end` pairs etc.) In doing this, we only need to move a single line (don't forget to reLaTeX each time!) and will get an estimate in which block the error is. Now delete everything from just after `"\begin{document}"` up to just before the suspicious block, and if the problem still remains, that is where the problem is.

3.1.2 Composite files

Basically, our procedure is similar to the procedure outlines above, just the end of a file that is read using `\input` is signified by `\endinput`. Beforehand, though, you should identify the parts responsible by successive commenting out of `\input` and/or `include` commands or by using `\includeonly`.

3.2 Simplifying

When the block containing the problem has been found, it needs to be simplified sufficiently, down to tracking down the bug to a single line. This is what the following examples will illustrate.

3.2.1 Removing unneccessary packages

There are packages that just won't work together. We will remove the packages step by step to see if this is our problem. To facilitate this, it is advisable to use the `\usepackage` command like this:

```
\usepackage{
  amsmath,
  listings,
  lmodern
}
```

so that packages can be commented out using a single %. As soon as the error disappears, re-enable the last package commented out and move on to the next.

3.2.2 Own commands and environments

User-defined commands and environments should be

- omitted completely if they are not used anymore in the remaining document,
- "emptied" otherwise, i.e. just have a #1 as second argument or do nothing at all.

This as well should be carried out step-by-step, in case the error is caused by one of these commands.

3.2.3 Shrinking

Now, we will basically go back to "moving the end", only not block- but linewise. I don't think this needs to be spelled out in full detail. Optimally, a single line of the document remains, that can be investigated further, or, if all else fails, posted to `comp.text.tex` (or suitable equivalents such as `de.comp.text.tex` or `fr.comp.tex`).

There are yet more ways of disabling parts of the document, such as the comment package, conditionals (if) and others.

3.2.4 Removing pictures

Insertion of pictures is (almost) always a problem. We would need to pass them along in order to get a compilable document. But they are usually of big file size and might contain confidental information. Hence, we cannot put them up on the web somewhere either. What do we do, then? Well, we can just replace the picture with a black rectangle of appropriate size, which we can generate using `\rule`.

3.2.5 Needed additional files

Sometimes there is no way round additional files to describe a problem, e. g. a small BibTEX file. LATEX offers a simple way to keep everything in one file and on demand the additional files are created on-the-fly with a given content: the `filecontents` environment.

```
1 \begin{filecontents}{filename.bib}
2   Some nonsense
3 \end{filecontents}
4 \documentclass[a4paper]{article}
5 ...
```

4 Examples

Please note: line numbers have been inserted for clarification and ease of reference. As given below, the examples are not LATEXable. Please omit the line numbers in your own examples.

4.1 Malformed Syntax

The following is a simple aligned equation, that could be created using the eqnarray environment, but the amsmath package is more powerful by far.

```
\documentclass[a4paper]{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
  S_1 > S > S_2 \\
  \Rightarrow S_1 & = \mathrm{e}^{}{x}  \\
  \Rightarrow S_1 & = \mathrm{e}^{}{y} \\
  \Rightarrow S_1 & = \mathrm{e^{}{x} \\
             & = \boxed{\mathrm{e}^{}{z}}
\end{align*}
\end{document}
```

The error message is:

```
Runaway argument

 S_1 > S > S_2 \\ \Rightarrow S_1 & = \mathrm {e}^{x} \\ \
    Rightarrow S_1 & =
\mathrm \ETC

 ! File ended while scanning use of \align*.
 <inserted text>
                 \par
```

The above leads us to suspect, that the error occurs in lines 6 to 9. Hence, we comment out all remaining lines in the align environment.

```
\documentclass[a4paper]{article}
\usepackage{amsmath}

\begin{document}
\begin{align*}
  S_1 > S > S_2 \\
%  \Rightarrow S_1 & = \mathrm{e}^{x}  \\
%  \Rightarrow S_1 & = \mathrm{e}^{y} \\
%  \Rightarrow S_1 & = \mathrm{e^{x} \\
%             & = \boxed{\mathrm{e}^{z}}
\end{align*}
\end{document}
```

The error does not occur anymore, indicating that we have removed too much. Reinserting the lines one by one, LaTeXing again after each one, shows that with line 8, the error reoccurs. Then, everything prior to this up to `\begin{align*}` is commented out, but the error still remains. Hence, we now delete those lines, resulting in an MWE, since the only package loaded is the one providing the align environment in the first place:

```
\documentclass[a4paper]{article}
\usepackage{amsmath}

\begin{document}
\begin{align*}
  \Rightarrow S_1 & = \mathrm{e^{x}
\end{align*}
\end{document}
```

A close look (and maybe help by your editor—any good editor can highlight matching parentheses) now reveals that we have forgotten the closing brace in `\mathrm{e^{x}`.

4.2 Undesired behaviour

Let us assume the following source:

```
\documentclass[a4paper]{article}
\usepackage{amsmath}
\newcommand{\im}{\mathrm{i}}
\newcommand{\e}{\mathrm{e}}
\begin{document}
\begin{align}
 f_N(t) = & \frac{A_0}{2} + \sum_{k=1}^{}\infty \left(  \frac
     {1}{2} \left( A_k -\im B_k \right) \e^{} {\im   \alpha t} + \
     frac{1}{2} \left( A_k +  \im B_k \right)   \e ^{} {- \im \
     alpha t} \right) \\
           & \text{mit } B_0=0 \text{ und } \alpha = \omega t
\end{align}
\end{document}
```

We dislike the apparently too small space in between the fraction and the equals sign created in line 7. We would like to detail this to the readers of `comp.text.tex`, but we are aware that this example is anything but clear. Hence, we limit the example to the relevant parts, possibly not including the user-defined commands in lines 3 and 4, which we verify by emptying them out.

```
\documentclass[a4paper]{article}
\usepackage{amsmath}
\newcommand{\im}{}
\newcommand{\e}{}
\begin{document}
\begin{align}
 f_N(t) = & \frac{A_0}{2} + \sum_{k=1}^{}\infty \left(
  \frac{1}{2} \left( A_k -\im B_k \right) \e^{} {\im
  \alpha t} + \frac{1}{2} \left( A_k +  \im B_k \right)
  \e^{} {- \im \alpha t} \right) \\
```

```
11          & \text{mit } B_0=0 \text{ und } \alpha
12             = \omega t
13 \end{align}
14 \end{document}
```

We are not interested in the correctness of the mathematics involved for our purpose. Yet our document is still quite complex (well, we will pretend this for the moment, since otherwise, this all makes little sense.) What we really need is line 7, and of this only the beginning, so we remove the rest and simplify some more.

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3
4 \begin{document}
5 \begin{align}
6  f = & \frac{a}{b}
7 \end{align}
8 \end{document}
```

And now, we have an MWE that illustrates our problem without lugging around additional overhead. The following might be suggested to fix problem:

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3
4 \begin{document}
5 \begin{align}
6   f & = \frac{a}{b}
7 \end{align}
8 \end{document}
```

4.3 Unexplainable behaviour ;-)

This example serves to illustrate mainly how to replace things that cannot be easily posted to the newsgroup as text. What is this to mean? When you insert pictures into your document, these are usually quite large and can usually cannot be embedded as text into a posting, since most servers do not allow binary messages in text groups such as `comp.text.tex`.

So we have a nice little document containing a picture and some text, and since we want to do a bit of maths, we have some additional packages for that.

```
1 \documentclass[a4paper,12pt]{scrreprt}
2 \usepackage{graphicx}
3 \usepackage{ngerman,amsmath,exscale}
```

```
4 \usepackage[latin1]{inputenc}
5 \begin{document}
6 Dies ist ein Testtext, der nur ein bisschen helfen soll,
    Neuankömmlingen in de.comp.text.tex das Prinzip des
    Minmalbeispiels zu erklären. Und wir fügen jetzt einfach mal
    ein Bild ein.
7 \includegraphics{bild} Und wir schreiben neben dem    Bild
    einfach weiter. Leider ist die Oberkante nicht    bündig mit
    dem folgenden Text, sowas dummes aber    auch. Da sollte man
    doch glatt ein Minimalbeispiel
8     basteln und in de.comp.text.tex nachfragen, warum    das so
        ist. Aber erst nach einer gründlichen
9     Eigenrecherche.
10 \end{document}
```

At first we check if the problem is related to any other package used, but commenting out line 3 does not change the result. Since we do not use a standard class here, but a replacement from the very recommendable KOMA-Script collection, we blame this class. Replacing `"scrreprt"` with `"report"` does not, however, improve the matter, but we leave this situation, since we can expect the report class on every installation. Now we reduce the remaining clutter and arrive at a truly minimal state.

```
1 \documentclass{report}
2 \usepackage{graphicx}
3
4 \begin{document}
5 \includegraphics{pic} Testtext
6 \end{document}
```

But, come to think of it, it is not truly minimal yet, so we replace the graphic. Now, we're done:

```
1 \documentclass{report}
2 %\usepackage{graphicx}
3
4 \begin{document}
5 \rule{3cm}{4cm} Testtext
6 \end{document}
```

4.4 Stumbling blocks apart from the source code

4.4.1 Trouble with frontends

Many people will use one of the common frontends or TEX-supporting editors (Emacs, Kile, TEXnicCenter, WinEdt to name but a few, with no claim to

completeness nor recommendation) to edit their files and the interface to LaTeX they provide (menu items, buttons, keyboard shortcuts to run LaTeX, BibTeX, etc.). But sometimes, even these programs can be misguided in the assistance they offer: they can be misconfigured, maybe they don't find some executables, maybe they don't spot error messages under some circumstances, maybe they are limited in their abilities. Maybe they think your paper is in ISO A5 format because you insert the fifth appendix via `\input{a5}`. If things are going wrong and you really think they shouldn't, keep the following in mind:

- If you're dealing with rotation, graphics or color, your dvi viewer will possibly not display these correctly. What counts is the ps or pdf output.
- Try to run `latex` or `pdflatex`, `bibtex`, `makeindex` etc. manually, i. e. from a command prompt or shell window. Maybe your editor is hiding their messages from you. `makeindex` is good at generating hard-to-spot errors:

```
This is makeindex, version 2.14 [02-Oct-2002] (kpathsea
  + Thai support).
Scanning style file ./svind.ist....done (4 attributes
  redefined, 0 ignored).
Scanning input file Approximative.idx....done (152 entries
  accepted, 2 rejected).
Sorting entries....done (1274 comparisons).
Generating output file Approximative.ind....done (210 lines
  written, 0 warnings).
Output written in Approximative.ind.
Transcript written in Approximative.ilg.
```

  Can you see it? Would you see it if it's in between two LaTeX runs that generate 120 lines of output each? Running one step at a time gives you time to check the output without having to read the log file, which can often be confusingly verbose.

4.4.2 Problems that won't go away

Sometimes, LaTeX or some package gets confused about the contents of auxiliary files. For example, try the following:

```
\documentclass{article}
\begin{document}
\makeatletter
\immediate\write\@auxout{foo}
\end{document}
```

You don't need to understand what's going on here beyond the fact it's writing "foo" to the aux file (go check!). The first run will succeed, the second run will fail, because you're not allowed to output text at the point when the aux file is read. If you abort `latex` (with x), no new aux file is generated and further runs will fail, too. This is completely independent of what you change the tex file to!

To narrow down where it's happening, you can count braces: when TeX opens a file, it puts `(filename` to the screen, and `)` when it closes them. In our case, we obtain something like

```
: ~ [0 514]; latex test
This is pdfeTeXk, Version 3.141592-1.20a-2.2 (Web2C 7.5.3)
 %&-line parsing enabled.
output format initialized to DVI
entering extended mode
(./test.tex
LaTeX2e &lt;2003/12/01&gt;
Babel &lt;v3.8d&gt; and hyphenation patterns for english,
dumylang, nohyphenation, german, ngerman, ukenglish, loaded.
(/opt/TL2004/texmf-dist/tex/latex/base/article.cls
Document Class: article 2004/02/16 v1.4f Standard LaTeX
document class (/opt/TL2004/texmf-dist/tex/latex/base/size10.clo)
    )
(./test.aux

! LaTeX Error: Missing \begin{document}.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.2 f
     oo
? x
No pages of output.
Transcript written on test.log.
: ~ [1 515];
```

Note that test.aux is the last file opened and not closed, so it's likely the error is there. (Well, I've already told you it's there.)

Of course, you can go the easy way by deleting the auxiliary files: we list some common extensions of auxiliary files along with where their content comes from and what it's for (so you know how to regenerate them after you delete them).

- aux: Labels, references, citations, babel settings, various other. Source:

tex or ltx file

- toc, lof, lot: Table of contents, lists of figures and tables. Source: tex or ltx file
- bbl: Bibliography. Source: bib and aux files via bibtex
- ind, idx: Index. Source: ind from idx via makeindex or xindy, idx from tex or ltx file
- gls, glo: Glossary. Just as ind and idx.