Correction sheets in LaTeX

Mike Piff

Abstract

In this article the author explains how to produce minor correction sheets to a LaTeX book. The sort of corrections handled are those which involve changes to a few words or possibly rewriting a few lines on a page, but not the sort that would cause TeX to reformat the whole book because a page has stretched or shrunk too much. The author has used these macros for corrections to a book in print, but they could equally well make the last stages of proof reading and correction less irksome.

1 Introduction

Reprinting individual pages of a LaTeX book for correction can be quite a tricky task. One can always process the whole book again, but chances are that the page breaks will all be different. This can also cause cross-references and the index to be inaccurate, and so is not really suitable unless the whole book is being reset.

The following style option `corrects.sty` makes the job easier. It was developed in particular for the author's needs, but should be easy to customize.

2 The style in use

It is assumed that a root file is available to produce the book. Make a copy of this, and insert the correction style option, and also the commands

```
\correctionsonly
\renewcommand{\resetcounters}[2]{%
   \setcounter{Theorem}{#1}%
   \setcounter{Example}{#2}%
   \ignorespaces}
```

in the preamble. The former is to tell LaTeX to produce only correction sheets, and the latter is customized to the particular counters being used in the current book—in this case, theorem and example counters. We must either make a copy of the `aux` file from the original book, or carefully replace any `\ref` and `\pageref` commands with their expansions in the corrections file. The author's own preference was the latter; he changed his root file so that it actually contained the pages to be printed, rather than have to `\input` or `\include` copies of each chapter file and `aux` file. As a precaution, he also redefined `\ref` and `\pageref` so as to give an error message.

Most of the body of the book may now be discarded. In its place, we put instructions to print individual pages. At least the text of these pages must be retained. But this is where the complications begin, since it is possible to be heavily nested within several layers of environments at the start of a page, as well as being mid-paragraph in section 9 of chapter 4, and about to produce Example 4.97 and Theorem 4.25. Moreover, at the end of the page we may be in a different paragraph and a different section. The headers on the pages will have to reflect the current state, and the page must finish flush right.

None of these problems is of much significance in itself, but the combination of all of them means that we have to be pretty careful in giving the exact state of the book for each page. Some counters change rapidly from page to page, whereas some change more slowly. I have divided them accordingly, and provide explicit commands to reset the slow counters, such as `chapter` and `section`, whilst allowing the individual page instruction to reset the quicker moving sort, such as `Theorem` or `Example`.

The instructions

```
\currentstate{chapter}{4}%
   {The theory of relaxors}
\currentstate{section}{9}%
   {Covariant relaxors}
```

tell LaTeX that we are firmly in chapter 4, section 9. LaTeX now needs to know how many theorems and examples are behind us. We will tell it at the start of the next page.

But first we need to inform LaTeX of the current state of nesting of environments at the top of the next page. We can do this either by leaving the immediately preceding text in, or by just giving it a clue like this.

```
\begin{enumerate}
   \item \mbox{}
   \item \mbox{}
   \item
      \begin{itemize}
         \item\mbox{}
         %%This is the text that
         %%immediately preceded the curr-
```

The paragraph is clearly in full flow, and so we tell LaTeX to start the page flush left, where "left" means according to the level of indentation of environments.

```
\startpageflushleft{101}{24}{96}%
   -ent page.
   Now we are producing page~101.
   The next theorem will be~25.
   The next example will be~97.
```

There is also a similar `\startpage` command to produce a normally indented paragraph at the top of the page, or to use if the first thing on the page is a theorem, say.

This page ends in mid-flow, so we let LaTeX know about this.

```
...it was clearly not too diff-%
\endpageflushright.
```

The `\endpage` command has a similar meaning, but allows normal paragraph termination in mid-line. Both cause the page to end flush bottom. Just use `\clearpage` if this page is at the end of a chapter.

It may happen that a run of two or more pages have mistakes on them. The macros in the style option are designed to make that easier to handle. At the start of a second, or subsequent, page in mid-paragraph, insert the instruction

```
\anotherpageflushleft
```

or include the command `\anotherpage` in its text if it does not have to start flush left. Terminate any such pages in the usual way. This saves having to retype the current state of the fast moving counters.

The author has found that floating figures are handled correctly, but TeX has to see the whole page where the figure is defined. Alternatively, the figure can be moved to an appropriate place on the page being printed. Clearly a large number of held-over insertions could present some problems, and the best course of action might then be to move them to the exact place where they should appear, with the "insert here" option active.

Footnotes on the current page can be handled by using the optional parameter to set the correct mark. Footnotes held over wholly from a previous page can be inserted with `\footnotetext`. A split footnote from a previous page is handled by means of the `\morefootnotetext` command, which takes the text of the footnote as its only parameter.

The final feature of this style option is the fact that *only* the pages specified above get through to the `dvi` file.

3 The style in detail

We first of all define two boolean variables. The variable `\ifcorrections` is an indicator of whether we are producing corrections or not. Its default is false, so the style will have no effect unless it is changed to true. Variable `\ifrealpage` is used internally to tell LaTeX to actually ship out the current page. Its default is true.

```
\newif\ifcorrections \correctionsfalse
\newif\ifrealpage \realpagetrue
```

To switch from this default mode, where the style has no effect, we provide a command to change the values of these two variables.

```
\def\correctionsonly{%
  \correctionstrue \realpagefalse}
```

The sectioning counters can be changed by means of the `\currentstate` command. We provide some dummy text by means of the `\mbox{}` command.

```
\def\currentstate#1#2#3{%
  \setcounter{#1}{#2}%
  \addtocounter{#1}{-1}%
  \csname #1\endcsname{#3}\mbox{}}%
```

To start a page of output, we specify the page number and then call a command `\resetcounters`, whose default meaning is to do nothing but ignore spaces. This command should be redefined in the user's preamble to take account of any counters that might need updating.

```
\def\startpage#1{\npage{#1}%
  \resetcounters}
\def\startpageflushleft#1{\npage{#1}%
  \noindent\resetcounters}
\newcommand{\resetcounters}{\ignorespaces}
```

Both commands to produce a new page make use of the `\npage` command.

```
\def\npage#1{\clearpage
  \global\realpagetrue
  \setcounter{page}{#1}}
```

The end of a page, flush right or not, is produced by the following two commands, or if appropriate by a `\clearpage`.

```
\def\endpageflushright{%
  {\parfillskip0pt\par\pagebreak}}
\def\endpage{\pagebreak}
```

If the page after this one also needs to be printed, an alternative way of producing it is to use one of the following two commands.

```
\def\anotherpageflushleft{%
  \global\realpagetrue\noindent}
\def\anotherpage{\global\realpagetrue}
```

A footnote from the previous page, part of which appears at the bottom of the current page, can be handled by inserting a left justified footnote without a mark in either the text or the footnote itself. This command should be invisible in the surrounding text, and so we make certain that it does not alter TeX' space factor.

```
\def\morefootnotetext#1{\@bsphack{%
  \def\@makefntext##1{\noindent ##1}%
  \footnotetext{#1}}\@esphack}
```

The only thing that remains is to make sure that any extraneous text, such as is created by sectioning commands, is not printed. We first take a copy of the LaTeX command that actually ships a page to the dvi file.

```
\let\@@outputpage\@outputpage
```

The command `\@outputpage` is then redefined in terms of its old meaning. First, only "real" pages are printed, that is, only when `\ifrealpage` returns true. Then, if we are only printing correction sheets, we immediately switch off printing of pages after this one. Only the `\startpage` and `\anotherpage` commands will switch printing on again.

Note that we need to ensure that any special page style set by `\thispagestyle` on a page that has not been printed does not carry over to the following page. For instance, a `\chapter` command will generally introduce a `plain` page style command into the document.

The command `\@outputpage` should otherwise do everything that its old version did, apart from shipping the page out to the dvi file.

```
\def\@outputpage{%
   \ifrealpage\@@outputpage
   \else\global\@specialpagefalse
   \let\firstmark\botmark\fi
   \ifcorrections\global\realpagefalse\fi}
```

◇ Mike Piff
Department of Pure Mathematics
University of Sheffield
Sheffield S10 2TN
England
Janet: M.Piff@shef.ac.uk

Text merges in TeX and LaTeX

Mike Piff

Abstract

In this article the author explains how to do some standard and not so standard word processor text merges in TeX documents, using no other tools than TeX itself. A common application is to the mail merge or form letter, where names and addresses are stored in a file, together with other bits of information, and a standard letter with variable fields embedded in it is customized for every name from this file. Another application is to the pretty-printing of the contents of a database.

The macros described in textmerg.sty work equally in both plain TeX and LaTeX.

1 Introduction

It is often said that although LaTeX is good at typesetting mathematics, it is wholly unsuitable for common word processor functions such as mail merges. The latter are easy to achieve in most ordinary word processors, but in its raw state LaTeX is incapable of doing a mail merge, or, indeed, of generating the same block of text over and over again but with different parameters in each block, those parameters having been read from a subsidiary merge file. The latter file might possibly be the output from a database or any other program.

This article aims to show the reader that such a repetitive task need not be as difficult as it at first appears. In TeX, it is possible to hide many details of a facility inside a subsidiary style file, so that the user is unaware of what fearful processes are going on in the background. It is then possible to present the end-user with an extremely simple interface, perhaps simpler and more powerful than is available in other systems.

In earlier *TUGboat* articles [Bel87, Gar87, Lee86, McK87] it was shown how a standard letter could be customized by adding names and addresses from a separate file. I aim to show that it is possible to achieve far more than this with a fairly compact but general set of macros.

2 A simple example

Suppose that we have a list of student names and examination grades, one per student, and that we wish to send a letter to each student giving his/her exam grade. We must decide first what bits of information must be prepared in our subsidiary file, by looking at an example letter and finding out which items change from letter to letter.

Suppose that one instance of our letter is the following, a LaTeX example.

```
\begin{letter}{Mr Abraham L Spriggs\\
   34 Winchester Road\\
   Sheffield S99 5BX\\
   England}
   \opening{Dear Mr Spriggs,}
   This letter is to inform you
   that you obtained grade C in
   your recent examinations.
   \closing{Yours faithfully,}
\end{letter}
```

We can see that we need to know the student's title, forename(s), surname, address and grade to compose such a letter.

One of the simplest ways of achieving this effect is to prepare a file with lines of the form