

The pdfreview package

Michael Palmer (mpalmer@uwaterloo.ca)

v1.2 (February 22, 2019)

Abstract

The `pdfreview` package lets you add comments in the page margins of PDF files, e.g. when reviewing manuscripts or grading reports. The PDF file to be annotated is included, one page at a time, as graphics, in a manner similar to the `pdfpages` package. Notes are placed in the margin next to the included graphics using a grid of guides. Alternatively, only numbers are placed in the page margins, and all the notes are collected into a numbered list at the end of the document.

Note that this package is *not* intended for adding notes into the L^AT_EX source of the document that is being reviewed; instead, the document undergoing review is already in PDF format and remains unchanged. Also note that this package does not produce the usual PDF ‘sticky notes’ that must be opened in the viewer by clicking on them; instead, the notes are simply shown as text in the page margin.

1 An example

Let’s assume a student submitted his essay as `lorem ipsum.pdf`. In order to review and annotate it, I create this wrapper L^AT_EX document:

```
\documentclass[letterpaper,10pt]{article}
\usepackage[
  sourcedoc={lorem ipsum.pdf},      % declare the PDF document being reviewed
  grid=true,                        % draw guides for margin notes (set false when done)
  staggered=true,                   % avoid vertical overlap of adjacent notes
  bodywidth=0.66,                   % set width of source doc page as fraction of \textwidth
  twocolumn=true,                   % enable margin notes on both sides
  trim=2cm,                          % trim all page margins of source doc by this amount
]{pdfreview}

\begin{document}

\begin{page}{1}                     % mount the first page

\begin{leftnotes}                   % write notes into the left margin
\cnote{84}{It looks like Latin.}    % place note vertically centered on guide 84
\bnote{68}{Is this really Latin?}  % align bottom of note to guide 68
\tnote{16}{No, not Latin}           % align top of note to guide 16
\end{leftnotes}                     % notes after this point go into the right margin

\tnote{48}{You really should be writing proper Latin}
\cnote{12}{Cite your references}
\end{page}

% \begin{page}{2}                     % mount the next page
% \note{...}{...} etc.

\end{document}
```

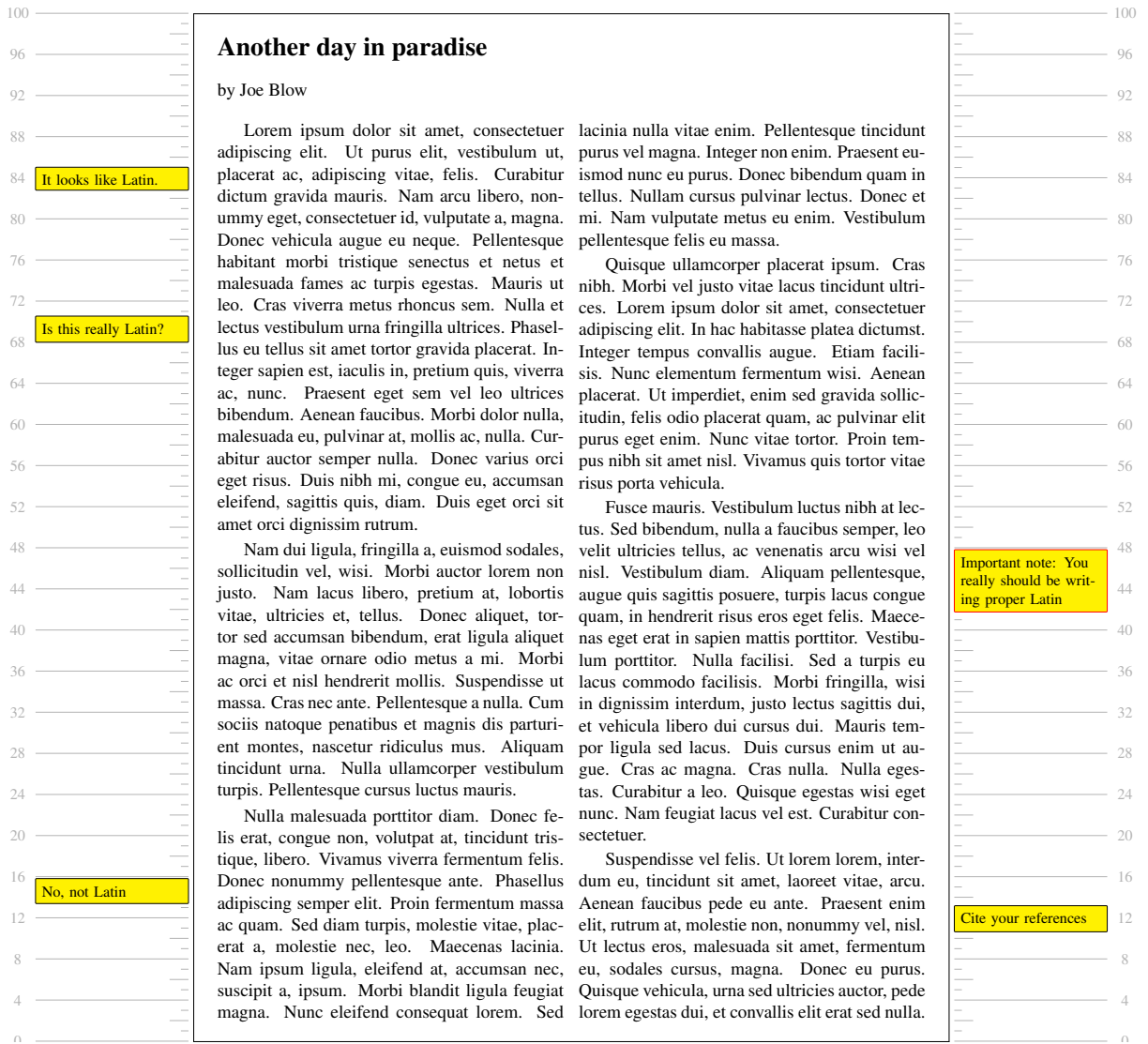


Figure 1: First page of the output produced by the sample listing above.

The first page of the output is shown in Figure 1. When you have added all your comments, you can set the package option `grid=false` in order to hide the guides in the page margins. That is already the gist of it; some more options and a few tips and tricks will be given below.

2 Package options

2.1 Options for controlling the selection and formatting of the source document

sourcedoc (*<file name>*; default: `noname`). Name of the PDF file to be reviewed. The file extension (`.pdf`) can be omitted. If the file name of your source document contains spaces, be sure to include it in braces.

bodywidth (*<number>*; default: `0.75`). Width of the included source doc page, as fraction of `\textwidth` of the wrapper document. The remainder of `\textwidth` will be used for the margin notes.

The default of 0.75 should be suitable in combination with `inline=true,twocolumn=false`, that is, with full note text in the right margin only (which is my own most common use case). Use smaller values with `inline=true,twocolumn=true` or larger ones with `inline=false`.

trim (*(length(s))*; default: 1cm). How much to trim off the margins of each page of the source document. This option is passed to the `\adjustbox` macro from the eponymous package. Accordingly, you can use two lengths to specify separate trimmings for horizontal and vertical margins, and four lengths to specify separate values for the left, lower, right, and top margins (in this order).¹

Note that multiple lengths values must be surrounded with braces, e.g. `trim={1cm 2cm}` or `trim={1cm 2cm 0cm 1cm}`. If you omit the braces, only the first length value will be used, and the others will be silently ignored.

2.2 Options for controlling the placement of notes

twocolumn (*(boolean)*; default: false). If true, place included page in the middle and arrange for margin notes on both sides. If false, place included page on the left and have margin notes on the right hand side only.

inline (*(boolean)*; default: true). If true, print the note text into the page margin. If false, print only the number of each note into the page margin. In the latter case, the note text will be collected and printed out as an `enumerate`'d list at the end of the document. The items in this list and the corresponding numbers in the margin will be reciprocally hyperlinked.

notenumbers (*(boolean)*; default: false). If true, start each note with its number. If option `inline` is false, the number is printed regardless of this option.

alignnotes (*(t, b, or c)*; default: c). Vertical alignment of a margin note declared with `\note`. Can be overridden by passing `t` or `b` as an optional argument (e.g. `\note[b]{Go boil your bottoms}` or `\note[t]{I shall taunt you a second time}`).

notesep (*(length)*; default: 3pt). Empty space between included page and margin notes.

staggered (*(boolean)*; default: true). If true, vertical overlap of one margin note by a second is avoided by pushing the second note downward below the first one. See section 5.2 for additional discussion.

maxscale (*(number)*; default: 100). Number of vertical length units/guides for placing margin notes—the height of the source doc page will be divided into `maxscale` units. If not divisible by 4, you will get a few ugly surplus lines at the top.

2.3 Options for controlling the appearance and formatting of notes and helplines

notesbg (*(color name or 'none')*; default: yellow). Background color to use for the margin notes. If 'none', the background will be transparent.

notesframe (*(color name or 'none')*; default: black). Color to use for the box frame of the margin notes. If 'none', no box is drawn.

important (*(frame color name)*; default: red). Color to use for frame around notes declared with style 'important' (e.g. `\note[important]{Important note!}`). Note that `important` is just a TikZ style declared inside the package; you can override this style with `\tikzset` take control of font, background color etc.

grid (*(boolean)*; default: true). Draw numbered guides for margin notes.

gridcolor (*(color name)*; default: black!30). Color to use for help guides and their numbers.

fontsize (*(string)*; default: footnotesize). Font size to use for the margin notes (and also the numbers on the guides, if present). Use any of the usual font size commands, but without the backslash.

¹No, this order doesn't resemble a clock, fuel gauge, or anything else that mere mortals might have encountered.

2.4 Other options

withnotesonly (*boolean*); default: `false`). By default, `pdfreview` will output every page that was mounted with `\begin{page} ... \end{page}`, even if no notes were placed on the page. When you set `withnotesonly=true`, pages on which no notes were made will be omitted from the output. This can be useful to reduce file size.

Extra pages inserted with `\begin{insertpage} ... \end{insertpage}` will always be included in the output, regardless of whether or not they were left empty.

scaletopage (*number*); default: `1`). Page number of source document to use for scaling all pages. It sometimes happens that the first page of a PDF document has different proportions from the subsequent ones; in this case, you can use this option to scale to the second page (`scaletopage=2`) or another one.

insertpagemargin (*length*); default: `2.5cm`). If you have comments of a general nature, or ones that don't fit comfortably into the page margin, you can put them on separate pages insert one or more empty pages using the `insertpage` environment (see below).

By default, `pdfreview` leaves only very narrow empty page margins beyond the margin notes. For inserted text pages, it is more appropriate to use a wider margin; you can adjust it with this option.

pageoffset (*number*); default: `0`). This option only affects the PDF bookmarks for each page of the current document; it is intended for synchronizing the bookmarks with the page numbers of the source document.

By default, `pdfreview` simply numbers all included pages consecutively, starting with 1. If the source document has, say, 5 unnumbered (or Roman-numbered) pages preceding the proper page 1, you can declare this with `pageoffset=-5`; in this case, the first 5 pages will be given Roman numerals, and Arabic numbers will start (at 1) on the sixth physical page. Similarly, if the source document is missing some logical pages at the beginning, you can correct for this using a positive value for `pageoffset`.²

3 Macros

`\note[alignment]{height}{text}` The optional `alignment` argument is one of `b`, `c`, or `t`. The `height` argument names the guide to align to; `text` is the content of the note. (The `height` argument doesn't have to be an integral number. Also, it can fall outside of the displayed guide scale, within reason.

Note that the entire text of each note will go inside a `TikZ` node. This means you cannot use empty lines within a note to insert paragraphs. If you do need paragraphs, use the `\par` macro instead.

`\bnote{height}{text}` Shorthand for `\note[b]{height}{text}`. Analogous `\cnote` and `\tnote`.

`\sourcedoc{file name}` Set the file name of the PDF file whose pages are to be included. More commonly, this is done using the `sourcedoc` package option (see above). However, this macro can be used to switch to another source document halfway through.

`\pagegrid[tikz settings][steps]` Occasionally, you may want to place a note right on top of the source page instead of in the margin. To this end, the `page` environment can be used like a `tikzpicture` environment (see section 4). This macro assists by drawing a grid of horizontal and vertical guides over the source doc pages. If you leave the `tikz settings` empty, the lines will be drawn in the same style as the guides for the margin notes, but you can use e.g. `\pagegrid[red, thick]` to make them stand out more. The `steps` parameter controls the density of the grid. The number of guides in each dimension is given by $100/\langle steps \rangle$; the default is 10.

²Attentive students of Korean appliance makers' owner's manuals will recognize this pattern—the least useful features require the most long-winded explanations.

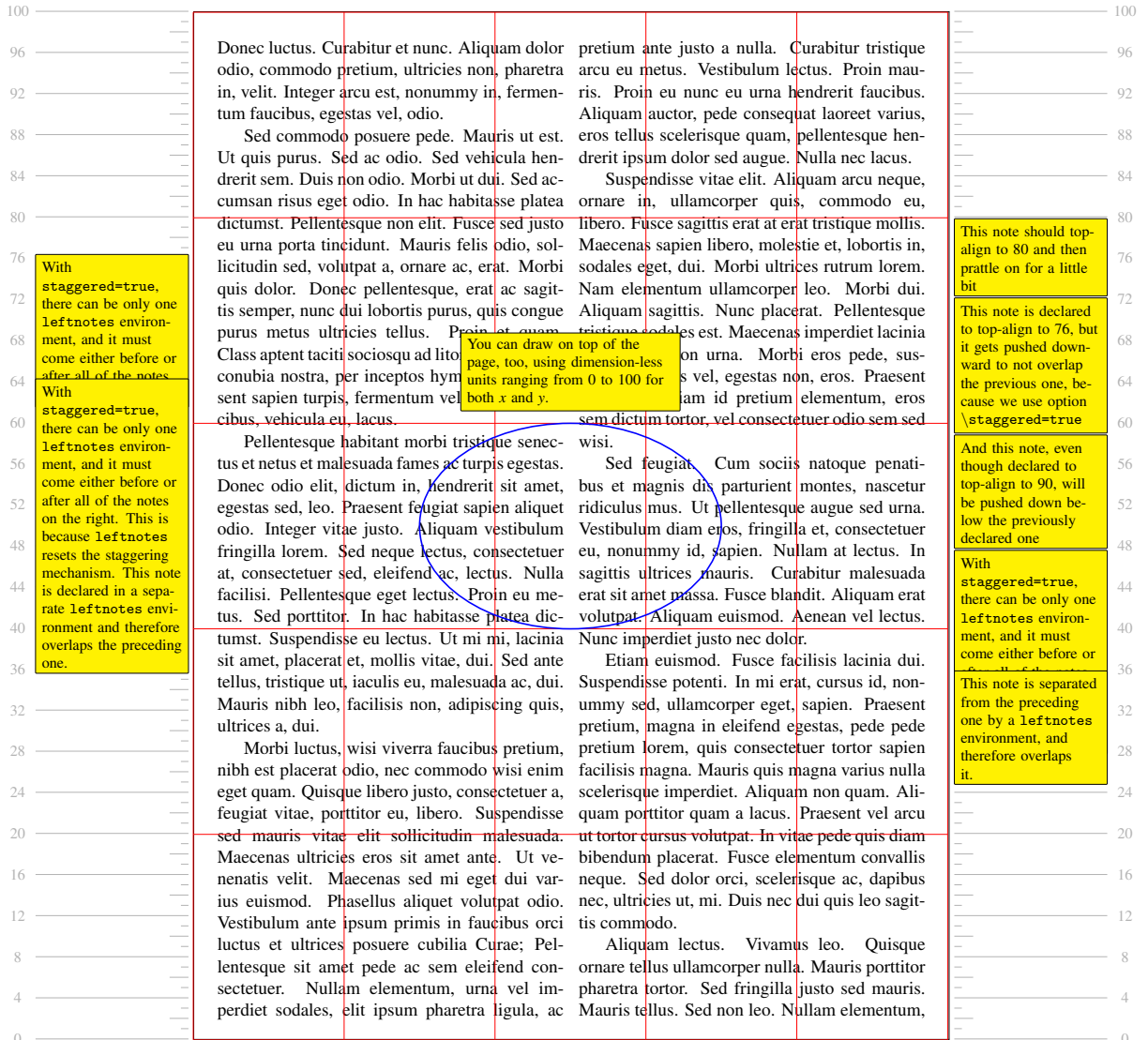


Figure 2: How to draw things on top of the included page, and how to muck up vertical spacing of the margin notes by misplacing a `leftnotes` environment.

4 Environments

`\begin{page}[\langle graphics options \rangle]{\langle page number \rangle} ... \end{page}` This environment loads the physical page $\langle page number \rangle$ from the source document and sets up the containers for the margin notes. The formatting and scaling of the included page and of the margin notes is subject to the package options described earlier.

The optional $\langle graphics options \rangle$ are passed to the `\adjustbox` macro from the package of the same name. You could use this for example to override the global `trim` package option for this one page, like so: `trim={0cm 1cm}`.

Another point to note is that the `page` environment can be used like a `tikzpicture` environment for drawing or placing nodes on top of the included source doc page. The dimension-less x and y units have been adjusted such that the point $(0,0)$ is at the bottom left, and $(100,100)$ is at the top right of the source doc page. That means you can say `\node[sticky] at (50,50) {\Sheesh!\}` to place some text smack-dab in the middle of the included page. (The `sticky` TikZ style will format the node

text like the notes in the margin.) You can use the `\pagegrid` macro to draw a grid of guides over the included page for this purpose (see section 3 and Figure 2).

`\begin{leftnotes}... \end{leftnotes}` This environment goes inside a `page` environment. In `twocolumn` mode, it will move the enclosed `\note` macros to the left page margin; with `twocolumn=false` it does nothing, that is, notes enclosed in `leftnotes` will also show up in the right margin. In this case, note that the `staggered=true` option will fail to prevent overlap between notes within and outside the `leftnotes` environment, respectively.

There should be only one `leftnotes` environment within each `page` environment, and it should come either before or after all notes intended for the right margin. Otherwise, the mechanism for avoiding vertical overlap of margin notes that is activated by the package option `staggered=true` will not work properly.

`\begin{rightnotes}... \end{rightnotes}` A no-op; it only exists to avoid tripping up users who might surmise its existence from that of the `leftnotes` environment.

`\begin{insertpage}[<heading>] ... \end{insertpage}` Inserts one or more extra pages for longer comments. These are not subject to any special formatting, except for the page margin, which can be set using the `insertpagemargin` package option (see section 2.4). Alternatively, you can set the margins manually using `\newgeometry` and `\restoregeometry` from the `geometry` package, which is loaded by `pdfpreview`. (The optional argument `<heading>` defaults to “General comments”.)

5 Practical recommendations

5.1 Page space usage

Obviously, one needs to make some judicious choices in order to allow enough space for our margin comments yet also keep the included PDF pages legible. Use the `trim` package option to shave off empty margins from the included PDF pages.

The `geometry` package will be loaded to divide up the page. By default, `pdfpreview` leaves 0.25 cm empty margins on both sides. If you load the `geometry` package yourself, `pdfpreview` will respect your settings and not impose its own margin width. In that case, note that the entire content—including PDF page and margin notes—will go inside `\textwidth`. Any margins you declare using the `geometry` package will remain entirely empty. Also note that any `geometry` settings can be declared in the preamble only; if you try to change the margins within the document using `\newgeometry`, things will unravel.

5.2 Vertical overlap of margin notes

Beginning with version 1.2, vertical overlap of margin notes can be prevented by using the package option `staggered=true`; this will cause any note that would overlap the preceding one to be pushed downwards (see Figure 2). I myself use it all the time, and I have therefore made this the default.

If for some reason you don’t want to use this convenience, then you will have to prevent vertical overlap manually. Where space is tight, it may help to use e.g. a `\note{60}{some note}` followed by a `\note{60}{another note}`; no space is wasted between both notes, yet at the same time they will not overlap.

The `bodywidth` package option also decides the horizontal and vertical space available for the margin notes. Should you decide half-way through to reduce the `bodywidth` parameter, this will create more horizontal space for the notes, but at the same time reduce the vertical space. This may occasionally create some new vertical overlap that would have to be fixed manually.

Another consideration is what font to use for your notes, and at what size. Times is a nice, space-saving font (in pdfLaTeX, you can load it with `\usepackage{mathptmx}`). You can adjust the font size using the corresponding package option (e.g. `fontsize=scriptsize`).

5.3 Writing less boiler plate code

While the code to set up each page is pretty compact, it can still become tiresome to write for long source documents. The package includes a `bash` script named `pdfreview.sh`, which generates a document skeleton containing a `page` environment for each page of a source document:

```
mpalmer@holzkopf:/data$ pdfreview.sh source.pdf > review.tex
```

The script requires 1) `ghostscript`, 2) `bash` (obviously) and 3) that you place it somewhere into your shell's path.

5.4 File size

Particularly when working with large source files, `pdfreview` can produce some startlingly large PDF output files. This problem is apparently related to the way `\includegraphics` works and has been described before (see <https://tex.stackexchange.com/questions/207527/>). It can often be amended by post-processing the output PDF file with `ghostscript`, for example:

```
gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dPDFSETTINGS=/printer -dNOPAUSE -dQUIET \
  -dBATCH -sOutputFile=post-processed.pdf output.pdf
```

The package contains a `bash` script named `pdfshrink.sh` that wraps the call to `ghostscript` and which you can use like

```
mpalmer@holzkopf:/data$ pdfshrink.sh large.pdf
```

An optional second argument controls the resolution of included bitmap graphics, with possible values (in order of decreasing resolution) `prepress`, `printer`, `default` (the default value), `ebook`, and `screen`. A optional third argument can be used to set the output file name.

Again, you need to save the script in a directory listed in your shell's search path for this work.

5.5 Uncooperative input files

Occasionally, I've run into source documents that failed to cooperate with the page-wise inclusion attempted by `pdfreview`, and/or with the `pdfreview.sh` code generation script, which internally uses `ghostscript`. I got around that by first printing the recalcitrant PDF input file to another PDF file—or, as long as `ghostscript` wasn't the problem, running through the `pdfshrink.sh` script—and then using the resulting file as the source document instead. (This might also be useful on occasion when using the `pdfpages` package.)

6 Implementation

This package depends on several others that are all in TeXLive. Among these is the `ulem` package, which makes the `\sout{strike me out}` command available for a strike-through effect. The `geometry` package is used for dividing up the page; if you load it yourself, your settings will be respected, as long you declare them in the preamble (but don't use `\newgeometry` halfway through the document). The `adjustbox` package is used to scale and trim the included pages, and also to scale the containers for the margin notes. The `environ` package is used to implement the `page` environment. The `tikz` package is used for placing guides and margin notes.

For further details, see the package file itself.

7 Revision history

May 2017 initial version (1.00)

September 2017 corrections to documentation; added `withnotesonly` option (1.1)

February 2019 (version 1.2)

- added package options `staggered`, `important`, and `scaletopage`
- removed `stretch` option
- added `\pagegrid` macro
- updated documentation
- updated auxiliary shell scripts to work with file names containing spaces, and to prevent loss of internal hyperlinks caused by ghostscript reprocessing

8 License

All files in this package are subject to the LaTeX project public license (see <http://www.latex-project.org/lppl.txt>).