# Page headers and footers in LaTeX

Piet van Oostrum*
Dept. of Computer Science
Utrecht University

January 3, 1995

**Abstract**

Standard LaTeX offers the page styles empty, plain, headings and myheadings. For simple page layouts these suffice, but they are rather rigid. A more flexible page layout may be obtained by using the fancyheadings package. This article describes how to create advanced page headers and footers in LaTeX documents by using this package. Some of the techniques described are also applicable with other page styles.

## Contents



*A considerable part of this article was written by George Grätzer (University of Manitoba) in *Notices Amer. Math. Soc.* Thanks, George!

# 1 What is fancyheadings

The fancyheadings macro package allows you to customize in LaTeX your page headers and footers in an easy way. You can define:

- three-part headers and footers
- decorative lines in headers and footers
- headers and footers wider than the width of the text
- multi-line headers and footers
- separate headers and footers for even and odd pages
- different headers and footers for chapter pages

Of course, you also have complete control over fonts, uppercase and lowercase displays, etc.

# 2 Where to get it?

You only need the file `fancyheadings.sty` which you can find at the CTAN sites: `ftp.shsu.edu` (U.S.), `ftp.tex.ac.uk` (U.K.), and `ftp.dante.de` (Germany). You will find it in the directory
`/tex-archive/macros/latex209/contrib/fancyheadings`. Although fancyheadings is a LaTeX 2.09 style file, it will work with LaTeX $2_\varepsilon$. Also this article and the two packages `fixmarks` and `extramarks` can be found there. (The code for these packages given in this article is a simplified version.)

# 3 Use of fancyheadings

To use this package in a LaTeX $2_\varepsilon$ document, place the file `fancyheadings.sty` in a directory/folder where TeX can find it (normally in the input directory/folder), and include in the preamble of your document after

```
\documentclass{...}
```

the commands:

```
\usepackage{fancyheadings}
\pagestyle{fancy}
```

We can visualize the page layout we can create with fancyheadings as follows:

| LeftHeader | CenteredHeader | RightHeader |
|---|---|---|
| | page body | |
| LeftFooter | CenteredFooter | RightFooter |

The LeftHeader and LeftFooter are left justified; the CenteredHeader and CenteredFooter are centered; the RightHeader and RightFooter are right justified.

We define each of the six "fields" and the two decorative lines separately.

## 4 A simple example

K. Grant is writing a report to Dean A. Smith, on "The performance of new graduates" with the following page layout:

**The performance of new graduates**

---

page body

---

| | | |
|---|---|---|
| From: K. Grant | To: Dean A. Smith | 3 |

where "3" is the page number. The title: "The performance of new graduates" is bold.

This is accomplished by these commands following `\pagestyle{fancy}`:

```
\lhead{}
\chead{}
\rhead{\bfseries The performance of new graduates}
\lfoot{From: K. Grant}
\cfoot{To: Dean A. Smith}
\rfoot{\thepage}
\setlength{\headrulewidth}{0.4pt}
\setlength{\footrulewidth}{0.4pt}
```

(The `\thepage` macro displays the current page number. `\bfseries` is the LaTeX $2_\varepsilon$'s way of selecting bold face.)

This is now fine, except that the first page does not need all these headers and footers. To eliminate all but the centered page number, issue the command

```
\thispagestyle{plain}
```

after the `\begin{document}` and the `\maketitle` commands.

Alternatively, issue

```
\thispagestyle{empty}
```

if you do not want any headers or footers.

In fact the standard LaTeX classes have the command `\maketitle` defined in such a way that a `\thispagestyle{plain}` is automatically issued. So if you *do* want the fancy layout on a page containing `\maketitle` you must issue a `\thispagestyle{fancy}` after the `\maketitle`.

## 5 An example of two-sided printing

Some document classes, such as `book.cls`, print two-sided: the even pages and the odd pages have different layouts; other document classes use the `twoside` option to print two-sided.

Now let us print the report two-sided. Let the above page layout be used for the odd (right-side) pages, and the following for the even (left-side) pages:

**The performance of new graduates**

---

page body

---

| | | |
|---|---|---|
| 4 | From: K. Grant | To: Dean A. Smith |

where "4" is the page number.

Here are the commands:

```
\lhead[]{\bfseries The performance of new graduates}
\chead{}
\rhead[\bfseries The performance of new graduates]{}
\lfoot[\thepage]{From: K. Grant}
\cfoot[From: K. Grant]{To: Dean A. Smith}
\rfoot[To: Dean A. Smith]{\thepage}
\setlength{\headrulewidth}{0.4pt}
\setlength{\footrulewidth}{0.4pt}
```

As you see, the values of the fields for the even (left-side) pages are given in the square brackets. Again, you may use `\thispagestyle{plain}` for a simple page layout for page 1.

## 6 Redefining plain style

Some LaTeX commands, like `\chapter`, use the `\thispagestyle` command to automatically switch to the `plain` page style, thus ignoring the page style currently in effect. To customize even such pages use the `fancyplain` page style. This page style sets up `fancy` for normal pages and in addition redefines the page style `plain` to conform more to the page style `fancy`. For instance, if all the page numbers are bold, you may want this on the first page also. You do this with the command

```
\pagestyle{fancyplain}
```

Note that this does not issue a `\pagestyle{plain}` or `\thispagestyle{plain}` command. This is either done by LaTeX (for instance, for the first page of a chapter in the `book` class), or you have to do it explicitly with the

```
\thispagestyle{plain}
```

command.

The redefinition of the plain style is done with the command

```
\fancyplain{plain_value}{normal_value}
```

inserted into all the appropriate arguments. As an example, let us redefine the plain style for the report in Section 5 by making the page number bold.

The leftheader definition:

```
\lhead[]{\bfseries The performance of new graduates}
```

changes to

```
\lhead[\fancyplain{}{}]{\fancyplain{}{\bfseries The
   performance of new graduates}}
```

Note that the first argument of both `\fancyplain` commands is empty, since the redefined plain style has an empty leftheader. (The `\fancyplain{}{}` is superfluous but for symmetry reasons it is given in this example.)

The left- and rightfooter definitions change more. For instance, the leftfooter definition changes from

```
\lfoot[\thepage]{From: K. Grant}
```

to

```
\lfoot[\fancyplain{}{\bfseries \thepage}]
      {\fancyplain{}{From: K. Grant}}
```

Carrying out these changes for all six fields, we get the code:

```
\lhead[\fancyplain{}{}]{\fancyplain{}{\bfseries The
   performance of new graduates}}
\chead{\fancyplain{}{}}
\rhead[\fancyplain{}{\bfseries The performance of new
   graduates}]{\fancyplain{}{}}
\lfoot[\fancyplain{}{\thepage}]{\fancyplain{}{From: K. Grant}}
\cfoot[\fancyplain{\bfseries \thepage}{From: K.
   Grant}]{\fancyplain{\bfseries \thepage}{To: Dean A. Smith}}
\rfoot[\fancyplain{}{To: Dean A. Smith}]{\fancyplain{}%
   {\bfseries \thepage}}
\setlength{\headrulewidth}{0.4pt}
\setlength{\footrulewidth}{0.4pt}
```

For the "fancy plain" page style, the thickness of the lines is defined by

`\plainheadrulewidth` and `\plainfootrulewidth`

(both default to 0pt). Observe that you may have different "fancy plain" page styles for even and odd pages.

To use the "fancy plain" page style, define these macros; in the preamble, include

```
\usepackage{fancyheadings}
\pagestyle{fancyplain}
```

and invoke the page style "fancy plain" with

```
\thispagestyle{plain}
```

where desired.

# 7 The default layout

Let us use the `book.cls` documentclass and the default settings for fancyheadings; so we only issue the commands

```
\usepackage{fancyheadings}
\pagestyle{fancy}
```

and let fancyheadings take care of everything. On the pages where new chapters start, we get a centered page number in the footer; there is no header, and there are no decorative lines.

On an even page, we get the layout:

| | |
|---|---|
| *1.2 EVALUATION* | *CHAPTER 1. INTRODUCTION* |

page body

4

On an odd page, we get the layout:

| | |
|---|---|
| *CHAPTER 1. INTRODUCTION* | *1.2 EVALUATION* |

page body

3

where the header text is slanted uppercase.

This default layout is produced by the following commands:

```
\lhead[\fancyplain{}{\slshape \rightmark}]{\fancyplain{}%
   {\slshape \leftmark}}
\chead{}
\rhead[\fancyplain{}{\slshape \leftmark}]{\fancyplain{}%
   {\slshape \rightmark}}
\lfoot{}
\cfoot{\rmfamily \thepage}
\rfoot{}
```

The following settings are used for the decorative lines:

| | |
|---|---|
| `\headrulewidth` | 0.4pt |
| `\footrulewidth` | 0 pt |
| `\plainheadrulewidth` | 0 pt |
| `\plainfootrulewidth` | 0 pt |

The header text is turned into all uppercase in `book.cls`.

## 8 The scoop on LaTeX's marks

Usually, for documents of class `book` and `report`, you may want to use chapter and section information in the headings (chapter only for one-sided printing), and for documents of class `article`, section and subsection information (section only for one-sided printing). LaTeX uses a marker mechanism to remember the chapter and section (section and subsection) information for a page; this is discussed in detail in the LaTeX *Companion*, Section 4.3.1.

There are two ways you can use and change the higher- and lower-level sectioning information available to you. The macros: `\leftmark` (higher-level) and `\rightmark` (lower-level) contain the information processed by LaTeX, and you can use them directly as shown in Section 7.

The `\leftmark` contains the left argument of the *last* `\markboth` on the page, the `\rightmark` contains the right argument of the *first* `\markboth` or the only argument of the *first* `\markright` on the page. If no marks are present on a page they are "inherited" from the previous page.

You can influence how chapter, section, and subsection information (only two of them!) is displayed by redefining the `\chaptermark`, `\sectionmark`, and `\subsectionmark` commands[1]. You must put the redefinition after the first call of `\pagestyle{fancy}` as this sets up the defaults.

Let us illustrate this with chapter info. It is made up of three parts:

- the number (say, 2), displayed by the macro `\thechapter`
- the name (in English, Chapter), displayed by the macro `\chaptername`
- the title, contained in the first argument of `\markboth`.

Figure 1 shows some variants for "Chapter 2. Do it now" (the last example is appropriate in some non-English languages).

For the lower-level sectioning information, do the same with `\markright`.

So if "Section 2.2. First steps" is the current section, then

```
\renewcommand{\sectionmark}[1]{
   \markright{\thesection.\ #1}}
```

will give "2.2. First steps"

Redefining the `\chaptermark` and `\sectionmark` commands may not eliminate all uppercaseness. E.g. the bibliography will have a title of BIBLIOGRAPHY in the header, as the `\uppercase` is explicitly given in the definition of `\thebibliography`. Similar for INDEX etc. If you don't want to redefine these commands, you can use an ugly hack like the following:

```
\lhead{\let\uppercase\relax\rightmark}
\rhead{\let\uppercase\relax\leftmark}
```

It should be noted that the LaTeX marking mechanism works fine with chapters (which always start on a new page) and sections (which are reasonably long). It does not work quite as well with short sections and subsections. This is a problem with LaTeX, not with fancyheadings.

[1] There are similar commands for `paragraph` and `subparagraph` but they are seldom used.

**Figure 1** Marker variants

| Code: | Prints: |
|---|---|
| `\renewcommand{\chaptermark}[1]{ \markboth{\chaptername\ \thechapter.\ #1}{}}` | Chapter 2. Do it now |
| `\renewcommand{\chaptermark}[1]{ \markboth{\uppercase{ \chaptername}\ \thechapter. \ #1}{}}` | CHAPTER 2. Do it now |
| `\renewcommand{\chaptermark}[1]{ \markboth{\uppercase{ \chaptername\ \thechapter. \ #1}}{}}` | CHAPTER 2. DO IT NOW |
| `\renewcommand{\chaptermark}[1]{ \markboth{#1}{}}` | Do it now |
| `\renewcommand{\chaptermark}[1]{ \markboth{\thechapter.\ #1}{}}` | 2. Do it now |
| `\renewcommand{\chaptermark}[1]{ \markboth{\thechapter.\ \chaptername.\ #1}{}}` | 2. Chapter. Do it now |

As an example let's take a page layout where the leftmarks are generated by the sections and the rightmarks by the subsections (as is default in the `article` class). Take a page with some short sections, e.g.
Section 1.
subsection 1.1
subsection 1.2
Section 2.

As the leftmark contains the *last* mark of the page it will be "Section 2.", and the rightmark will be "subsection 1.1" as it will be the *first* mark of the page. So the page header info will combine section 2 with subsection 1.1 which isn't very nice. The best you can do in these cases is use only the `\rightmarks` and redefine `\sectionmark` accordingly. A LaTeX command `\firstleftmark` would also be a nice addition (see the `extramarks` package in section 16).

Another problem with the marks in the standard LaTeX classes is that the higher level sectioning commands (e.g. `\chapter`) call `\markboth` with an empty right argument. This means that on the first page of a chapter (or a section in article style) the `\rightmark` will be empty. If this is a problem you must manually insert extra `\markright` commands or redefine the `\chaptermark` (`\sectionmark`) commands to issue a `\markboth` command with two decent parameters.

As a final remark you should also note that the * forms of the `\chapter` etc. commands do *not* call the mark commands. So if you want your preface to set the header info but not be numbered nor be put in the table of contents, you must issue the `\markboth` command yourself, e.g.

```
\chapter*{Preface\markboth{Preface}{}}
```

Entering the `\markboth` command inside the `\chapter*` insures that the mark will not be separated from the title by a page break. Of course with `\chapter` this wouldn't be a problem if you put the mark command after the chapter title, as the `\chapter` command starts at a new page. However with a `\section*` it could be dangerous to say:

```
\section*{Preface}
\markboth{Preface}{}
```

as a page break may occur between the two commands.

# 9 Dictionary style headers

Dictionaries and concordances usually have a header containing the first word defined on the page or both the first and the last words. This can easily be accomplished with fancyheadings and LaTeX's `mark` mechanism. Of course if you use the marks for dictionary style headers, you cannot use them for chapter and section information, so if there are also chapters and sections present, you must redefine the `\chaptermark` and `\sectionmark` to make them harmless:

```
\renewcommand{\chaptermark}[1]{}
\renewcommand{\sectionmark}[1]{}
```

Now you do a `\markboth{#1}{#1}` for each dictionary or concordance entry `#1` and use `\rightmark` for the first entry defined on the page and `\leftmark` for the last one.

If you want to use a header entry of the form firstword–lastword it would be nice if this would be reduced to just the form firstword if both are the same. This could happen if there is just one entry on the page. In this case a test must be made to check if the marks are the same. However, TeX's marks are strange beasts, which cannot be compared out of the box with the plain TeX `\if` commands. Fortunately the `ifthen` package works well:

```
\newcommand{\mymarks}{
  \ifthenelse{\equal{\leftmark}{\rightmark}}
    {\rightmark}
    {\rightmark--\leftmark}}
\lhead[\rm \mymarks]{\rm \thepage}
\rhead[\rm \thepage]{\rm \mymarks}
```

Dictionaries are often done with two columns. Unfortunately there is a bug in LaTeX's twocolumn option which causes some marks to be lost. If you use the package in figure 2, this will be solved[2][3].

[2] This is a simplification of the actual code. Get the real package from the CTAN sites.
[3] The `multicol` package uses a similar technique.

**Figure 2** Fix marks in twocolumn style

```
% fixmarks.sty:
% Patch LaTeX's output routine to handle marks correctly with two columns.
% Joe Pallas <pallas@edu.stanford.neon>
% Corrected by Piet van Oostrum <piet@cs.ruu.nl> on Feb 5, 1993, Oct 5, 1994

\def\@outputdblcol{\if@firstcolumn \global\@firstcolumnfalse
% Remember the marks from the first column
    \global\setbox\@leftcolumn\copy\@outputbox
    \splitmaxdepth=\maxdimen \cbaddness=10000
    \setbox\@outputbox\vsplit\@outputbox to\maxdimen
    \xdef\@firstcoltopmark{\topmark}%
    \xdef\@firstcolfirstmark{\splitfirstmark}%
    \ifx\@firstcolfirstmark\empty\global\let\@setmarks\relax\else
      \gdef\@setmarks{\let\firstmark\@firstcolfirstmark
        \let\topmark\@firstcoltopmark}%
    \fi
% End of change
  \else \global\@firstcolumntrue
    \setbox\@outputbox\vbox{\hbox to\textwidth{\hbox to\columnwidth
      {\box\@leftcolumn \hss}\hfil \vrule width\columnseprule\hfil
       \hbox to\columnwidth{\box\@outputbox \hss}}}\@combinedblfloats
% Override current first and top with those of first column if necessary
       \@setmarks
% End of change
       \@outputpage \begingroup \@dblfloatplacement \@startdblcolumn
       \@whilesw\if@fcolmade \fi{\@outputpage\@startdblcolumn}\endgroup
    \fi}
```

## 10 Fancy layouts

You can make a multi-line field with the `\\` command. It is also possible to put extra space in a field with the `\vspace` command. Note that if you do this you will probably have to increase the height of the header (`\headheight`) or of the footer (`\footskip`), otherwise you may get error messages "Overfull `\vbox` ...has occurred while `\output` is active". See Section 4.1 of the LaTeX *Companion* for detail.

For instance, the following code will place the section title and the subsection title of an article in two lines in the upper right hand corner:

```
\documentclass{article}
\usepackage{fancyheadings}
\pagestyle{fancy}
\addtolength{\headheight}{\baselineskip}
\renewcommand{\sectionmark}[1]{\markboth{#1}{}}
\renewcommand{\subsectionmark}[1]{\markright{#1}}
\rhead{\leftmark\\\rightmark}
```

You can customize the decorative lines. You can make the decorative line in the header quite thick with

```
\setlength{\headrulewidth}{0.6pt}
```

or you can make the decorative line in the footer disappear with

```
\setlength{\footrulewidth}{0pt}
```

The decorative lines, themselves, are defined in the two macros `\headrule` and `\footrule`. For instance, if you want a dotted line rather than a solid line in the header, redefine the command `\headrule`:

```
\renewcommand{\headrule}{\vbox to 0pt{\hbox
    to\headwidth{\dotfill}\vss}}
```

## 11 Two book examples

The following definitions give an approximation of the style used in L. Lamport's LaTeX book.

Lamport's header overhangs the outside margin. This is done as follows.

The width of headers and footers is `\headwidth`, which by default equals the width of the text: `\textwidth`. You can make the width wider (or narrower) by redefining `\headwidth` with the `\setlength` and `\addtolength` commands. To overhang the outside margin where the marginal notes are printed, add both `\marginparsep` and `\marginparwidth` to `\headwidth` with the commands:

```
\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}
```

You must issue these commands *after* the first `\pagestyle{fancy}` or `\pagestyle{fancyplain}` command as this will establish the default for `\headwidth`.

And now a complete definition of Lamport's book style:

```
\documentclass{book}
\usepackage{fancyheadings}
\pagestyle{fancyplain}
\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\lhead[\fancyplain{}{\bfseries\thepage}]
   {\fancyplain{}{\bfseries\rightmark}}
\rhead[\fancyplain{}{\bfseries\leftmark}]
   {\fancyplain{}{\bfseries\thepage}}
\cfoot{}
```

Notice that the `\chaptermark` and `\sectionmark` commands have been redefined to eliminate the chapter numbers and the uppercaseness.

For the second example, we take the AMS-LaTeX book.

Chapter pages have no headers or footers. So we declare

```
\thispagestyle{empty}
```

for every chapter page, and we do not need fancyplain.

Chapter and section titles appear in the form: 2. DO IT NOW, so we have to redefine `\chaptermark` and `\sectionmark` as follows (see Section 8):

```
\renewcommand{\chaptermark}[1]%
   {\markboth{\uppercase{\thechapter.\ #1}}{}}
\renewcommand{\sectionmark}[1]%
   {\markright{\uppercase{\thesection.\ #1}}}
```

In an even-header, the page number is printed as the LeftHeader and the chapter info as the RightHeader; in an odd-header, the section info is printed as the LeftHeader and the page number as the RightHeader. The CenteredHeaders are empty. There are no footers.

There is a decorative line in the header. It is 0.5pt wide, so we need the commands:

```
\setlength{\headrulewidth}{0.5pt}
\setlength{\footrulewidth}{0pt}
```

The font used in the headers is 9 pt bold Helvetica. The PSNFSS system by Sebastian Rahtz uses the short (Karl Berry) name `phv` for Helvetica, so this font is selected with the commands:

```
\fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont
```

(See Sections 7.6.1 and 11.9.1 of the LaTeX *Companion*.) Let us define a shorthand for this:

```
\newcommand{\helv}{%
   \fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont}
```

Now we are ready for the page layout:

```
\lhead[\helv \thepage]{\helv \rightmark}
\rhead[\helv \leftmark]{\helv \thepage}
\cfoot{}
```

Putting this all together:

```
\documentclass{book}
\usepackage{fancyheadings}
\pagestyle{fancy}
\renewcommand{\chaptermark}[1]%
   {\markboth{\uppercase{\thechapter.\ #1}}{}}
\renewcommand{\sectionmark}[1]%
   {\markright{\uppercase{\thesection.\ #1}}}
\setlength{\headrulewidth}{0.5pt}
\setlength{\footrulewidth}{0pt}
\newcommand{\helv}{%
   \fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont}
\lhead[\helv \thepage]{\helv \rightmark}
\rhead[\helv \leftmark]{\helv \thepage}
\cfoot{}
```

## 12 Special page layout for float pages

Some people want to have a special layout for float pages (pages only containing floats). As these pages are generated autonomically by LaTeX, the user doesn't have any control over them. There is no `\thispagestyle` for float pages and any change of the page style will at least also affect the page before the float page. With fancyheadings you can give the following definition in your preamble[4]:

```
\makeatletter
\def\iffloatpage#1#2{\if@fcolmade #1\else #2\fi}
\makeatother
```

In each of the header- or footer fields you can now specify
`\iffloatpage{value for float page}{value for other pages}`
You can even use this to get rid of the decorative line on float pages only by defining:

```
\renewcommand{\headrulewidth}{\iffloatpage{0pt}{0.4pt}}
```

Note that `\renewcommand` is used rather than the usual `\setlength` command to change the `\headrulewidth`. Changing it with `\setlength` will not work[5].

After this `\renewcommand` it will no longer be possible to change the parameter `\headrulewidth` with `\setlength`, you will have to use `\renewcommand` in the rest of the document.

Sometimes you may want to change the layout also for pages that contain a float on the top of the page or a float on the bottom of the page.

Enter the following definitions in a style file (or between `\makeatletter` and `\makeatother` in your preamble:

```
\let\latex@makecol\@makecol
\def\@makecol{\let\topfloat\@toplist\let\botfloat\@botlist\latex@makecol}
\def\iftopfloat#1#2{\ifx\topfloat\empty #2\else #1\fi}
\def\ifbotfloat#1#2{\ifx\botfloat\empty #2\else #1\fi}
```

You can then use the commands `\iftopfloat` and `\ifbotfloat` similar to `\iffloatpage`.

Note: Marks in floats will not be visible in LaTeX's output routine, so it is not useful to put marks in floats. So there is currently no way to let a float (e.g. a figure caption) influence the page header or footer.

## 13 Those blank pages

In the `book` class when the `openany` option is not given or in the `report` class when the `openright` option is given, chapters start at odd-numbered pages, half of the time causing a blank page to be inserted. Some people prefer this page to be completely empty, i.e. without headers and footers. This cannot be done

[4] If you have fancyheadings version 1.5 or higher the commands described in this section are already included

[5] It was probably a dumb decision to define the width parameters as lengths. There is no need to and I may change this in a future version of fancyheadings

with `\thispagestyle` as this command would have to be issued on the *previous* page. There is, however, no magic necessary to get this done:

```
\clearpage{\pagestyle{empty}\cleardoublepage}
```

As the `\pagestyle{empty}` is enclosed in a group it only affects the page that may be generated by the `\cleardoublepage`. You can of course put the above in a private command. If you want to have this done automatically at each chapter start or when you want some other text on the page then you must redefine the `\cleardoublepage` command.

```
\makeatletter
\def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
  \hbox{}
  \vspace*{\fill}
  \begin{center}
    This page intentionally contains only this sentence.
  \end{center}
  \vspace{\fill}
  \thispagestyle{empty}
  \newpage
  \if@twocolumn\hbox{}\newpage\fi\fi\fi}
\makeatother
```

## 14 N of M style page numbers

Some document writers prefer the pages to be numbered as n of m where m is the number of pages in the document. There is a package `nofm.sty` available, but some versions of it are defective, and most don't work with fancyheadings because they take over the complete page layout. For LaTeX 2$\varepsilon$ there is a package `lastpage` available which you can use with fancyheadings as follows:

```
\usepackage{lastpage}
...
\cfoot{\thepage\ of \pageref{LastPage}}
```

If you are still using LaTeX2.09 and you are not able to switch to LaTeX 2$\varepsilon$ you can define a LaTeX2.09 compatible `lastpage.sty` as follows:

```
\let\origenddocument=\enddocument
\def\enddocument{\clearpage\if@filesw
  {\addtocounter{page}{-1} \immediate\write\@mainaux
    {\string\newlabel{LastPage}{{}{\thepage}}}}\origenddocument}
```

With some trickery[6] the value of the `LastPage` label can be used to make different headers or footers on the last page of a document. E.g. if you want the footer of every odd page, except if it is the last one, to contain the text "please turn over", this can be done as follows:

[6]If you have a recent version of the `ifthen` package the tricks are not necessary. You can then test the `\pageref{LastPage}` directly.

```
\usepackage{lastpage}
\usepackage{ifthen}
...
\makeatletter
\def\LastPagenum{\@ifundefined{r@LastPage}{0}{\expandafter
    \expandafter\expandafter\@cdr
    \csname r@LastPage\endcsname\@nil\null} }
\makeatother
\rfoot{\ifthenelse{\isodd{\value{page}} \and \not
    \value{page}=\LastPagenum}{please turn over}{}}
```

## 15 When to change the headers and footers?

Sometimes you want to change the header or footer layout in the course of a document. Some of these changes can be accomplished by using the mark mechanism as may be seen in section 8 and 16. However, sometimes we want a more drastic change, e.g to change the page numbering from roman to arabic (with `\pagenumbering`), to change one of the fancyheadings fields or to change to another page style. Sometimes you may be surprised to find the change to occur too early. In general the above mentioned changes take effect immediately, i.e. on the page that is currently being built. If you want the change to take effect at the next page you must make sure that the current page is finished. In most cases this can be done by issuing a `\clearpage` command before any of the above mentioned changes. If this is not possible you can use the `afterpage` package with:
`\afterpage{\lhead{new value}}` or `\afterpage{\pagenumbering{roman}}`. You cannot use `\afterpage` to change the `\pagestyle` as the commands issued by `\afterpage` are local in a group, and the `\pagestyle` command makes only local changes. The `\pagenumbering` and the fancyheadings commands make global changes so they will work, as will the `\thispagestyle` command.

It should be noted that although the fancyheadings commands like `\lhead` take effect immediately, this does not mean that any "variables" used in these commands get the value they have at the place where these commands are given. E.g. if `\cfoot{\thepage}` is given the page number that will be inserted in the footer is not the page number of the page where this command is given, but rather the page number of the actual page where the footer is constructed. Of course for the page number this is what you expect, but it is also true for other commands.

So if you have a book where each chapter is written by a different author and you want the name of the author in the lower left-hand corner you can use the following commands:

```
\newcommand{\TheAuthor}{}
\newcommand{\Author}[1]{\renewcommand{\TheAuthor}{#1}}
\lfoot{\TheAuthor}
```

and start each chapter with the command `\Author{Real Name}`. If however, the author name would be changed before a page is completed the wrong author could come in the footer. This would be the case if you gave the above command *before* the `\chapter` command rather than after it. Another source of problems is the fact that TEX's output routine processes commands ahead, so it may already have processed some commands that produce text that will appear on the next page. See the next section for an example.

## 16 Headers and footers induced by the text

We have seen how we can use LATEX's marks to get information from the document contents to the headers and footers. The marks mechanism is the only reliable mechanism that you can use to get changing information to the headers or footers. This is because LATEX may be processing your document ahead before deciding to break the page.

Sometimes the two marks that LATEX offers are not enough. An example is the following:

> If a solution to an exercise goes across a page break, then I would like to have "(Continued on next page...)" at the bottom of the first page and "(Continued...)" at the top in the margin of the next page.

You cannot use LATEX's mark mechanisms for this if you also want to use chapter and section information.

The code from figure 3 constitutes a package that gives you two extra marks that can be used in this situation[7]. Here is a way to use this package:

```
\usepackage{extramarks}
...
\pagestyle{fancy}
\lhead{\firstxmark}
\rfoot{\lastxmark}
...
\extramarks{}{Continued on next page\ldots}
Some text that may or may not cross a page boundary...
\extramarks{Continued\ldots}{}
```

Note that the `\extramarks` command must be close to the text, i.e no empty lines (paragraph boundaries) should intervene. Otherwise the page may be broken at that boundary and the extramarks would come on the wrong page.

There are two new marks that can be used in the page layout with this package: If commands of the form `\extramarks{`$m_1$`}{`$m_2$`}` are given `\firstxmark` gives you the first $m_1$ value and `\lastxmark` gives you the last $m_2$ value of the current page. It also gives you the `\firstleftmark` and `\lastrightmark` commands that complement the standard LATEX marks.

To stress the point that marks are the correct way to do this, let me give you a "solution" that will not work[8]:

[7] After I made this package I discovered a package `secret.sty` that does a similar thing to mark confidential paragraphs if they cross a page boundary. It does it, however, by changing the output routine.

[8] Actually there is another way but it requires two LATEX passes: you can put `\label` commands before and after the text and compare the `\pageref`s.

**Figure 3** Package for extra marks in LaTeX

```
% extramarks.sty
\def\@leftmark#1#2#3#4{#1}
\def\@rightmark#1#2#3#4{#2}

\def\markboth#1#2{{\def\protect{\noexpand\protect\noexpand}
     \let\label\relax \let\index\relax \let\glossary\relax
     \expandafter\@markboth\@themark{#1}{#2}
     \mark{\@themark}}\if@nobreak\ifvmode\nobreak\fi\fi}
\def\markright#1{{\def\protect{\noexpand\protect\noexpand}
     \let\label\relax \let\index\relax \let\glossary\relax
     \expandafter\@markright\@themark
     {#1}\mark{\@themark}}\if@nobreak\ifvmode\nobreak\fi\fi}
\def\@markright#1#2#3#4#5{\gdef\@themark{{#1}{#5}{#3}{#4}}}
\def\@markboth#1#2#3#4#5#6{\gdef\@themark{{#5}{#6}{#3}{#4}}}
\def\leftmark{\expandafter\@leftmark\botmark{}{}{}{}}
\def\rightmark{\expandafter\@rightmark\firstmark{}{}{}{}}
\def\firstleftmark{\expandafter\@leftmark\firstmark{}{}{}{}}
\def\lastrightmark{\expandafter\@rightmark\botmark{}{}{}{}}

\def\@themark{{}{}{}{}}

\def\extramarks#1#2{{\def\protect{\noexpand\protect\noexpand}
     \let\label\relax \let\index\relax \let\glossary\relax
     \expandafter\@markextra\@themark{#1}{#2}
     \mark{\@themark}}\if@nobreak\ifvmode\nobreak\fi\fi}
\def\@markextra#1#2#3#4#5#6{\gdef\@themark{{#1}{#2}{#5}{#6}}}
\def\firstxmark{\expandafter\@firstxmark\firstmark{}{}{}{}}
\def\topxmark{\expandafter\@firstxmark\topmark{}{}{}{}}
\def\lastxmark{\expandafter\@lastxmark\botmark{}{}{}{}}
\def\@firstxmark#1#2#3#4{#3}
\def\@lastxmark#1#2#3#4{#4}
```

```
\lhead{Continued}
\rfoot{Continued on next page\ldots}
Some text that may or may not cross a page boundary...
\lhead{}
\rfoot{}
```

You may be tempted to think that the first `\lhead` and `\rfoot` will be in effect when TeX breaks the page in the middle of the text, and the last ones when the page breaks after the text. This is not true as the whole paragraph (including the last definitions) will be processed before TeX considers the page break, so at the time of the page break the last definitions are effective, whether the page break occurs inside the text or outside of it. Putting a paragraph boundary between the text and the last definitions will not work either, because you don't want the first definitions to be in effect when TeX decides to break the page exactly at this boundary. Actually the marks mechanism was invented to get rid of these kinds of problems.

In the above example the text "Continued" appears in the page header. It may be nicer to put it in the margin. This can be easily accomplished by positioning it at a fixed place relative to the page header. In plain TEX you would use a concoction of `\hbox to 0pt`, `\vbox to 0pt`, `\hskip`,`\vskip`, `\hss` and `\vss` but fortunately LATEX's `picture` environment gives a much cleaner way to do this. In order not to disturb the normal header layout we put the text in a zero-sized `picture`. Generally this is the best way to position things on fixed places on the page. You can then also use the normal headings. See also section 18 for another example of this technique.

```
\setlength{\unitlength}{\baselineskip}
\lhead{\begin{picture}(0,0)
  \put(-2,-3){\makebox(0,0)[r]{\firstxmark}}
  \end{picture}\leftmark}
```

This solution can of course also be used for the footer. Be sure to put the `picture` as the first thing in left-handside entries and last in right-handside ones.

Finally you may want to put "(Continued...)" in the *text* rather than in the header or the margin. Then you have to use the `afterpage` package. We also decide to make a separate environment for it.

```
\newenvironment{continued}{\par
  \extramarks{}{Continued on next page\ldots}
  \afterpage{\noindent\firstxmark\vspace{1ex}}
  }{\extramarks{(Continued\ldots)}{}\par}
```

It is a bit dangerous to use `\firstxmark` outside the page layout routine, but apparently with `\afterpage` this works. If you would need the information further on in the page you must remember the state of the marks in your own variable. You can do this in one of the fancyheadings fields. For example if you want to add something *after* the broken piece of text you can use the following:

```
\newcommand{\mysaved}{}

\newenvironment{continued}{\par
  \extramarks{}{Continued on next page\ldots}
  }{\extramarks{(Continued\ldots)}{}\par\vspace{1ex}\mysaved}
\lhead{\leftmark}
\chead{\ifthenelse{\equal{\lastxmark}{}}
  {\gdef\mysaved{}}
  {\gdef\mysaved{\noindent[Continued from previous page]}}}
```

If you want to include one of the marks or other varying information in the saved text, you must use `\xdef` rather than `\gdef`.

# 17 A movie

If you put at each page on the same place a picture that slightly changes from page to page you can get a movie-like effect by flipping through the pages. You can create such a movie easily with fancyheadings. For simplicity we assume that each picture is in a postscript (EPS) file called `pic`$\langle n \rangle$`.ps` where $\langle n \rangle$ is the page number and that we use the `epsf` package. To put the movie in the righthandside bottom corner the following will work:

```
\rfoot{\setlength{\unitlength}{1mm}\setlength{\epsfxsize}{2cm}
  \begin{picture}(0,0)
    \put(5,0){\epsfbox{pic\thepage.ps}}
  \end{picture}}
```

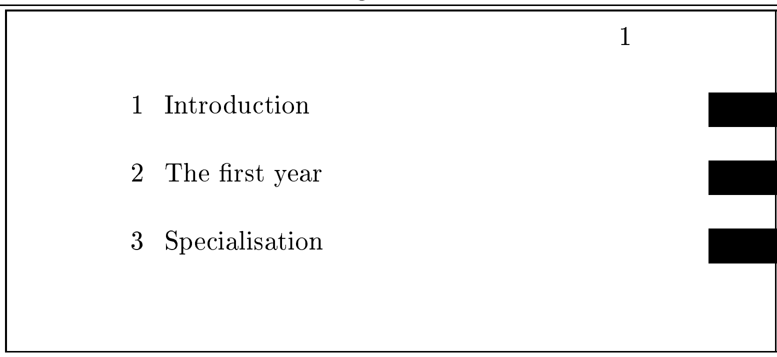
Notice that the `\unitlength` and `\epsfxsize` parameter should be set locally in the fancyheadings field in order to avoid unwanted interference from their values in the text.

## 18 Thumb-indexes

Some railroad guides and expensive bibles have so called *thumb-indexes*, i.e. there are marks on the sides of the pages that indicate where the chapters are. You can create these by printing black blobs in the margin of the pages. The vertical position should be determined by the chapter number or some other counter. As the position is independent of the contents of the page, we print these blobs as part of the header in a zero-sized `picture` as described in the previous section.

Of course we have to take care of two-sided printing, and we may want to have an index page with all the blobs in the correct position. The solution requires some hand-tuning to get the blobs nicely spaced out vertically. For the application that I had there were 12 sections, so I made the blobs 18 mm apart, i.e. 9 mm blob separated by 9 mm whitespace. In order to avoid calculations they are set in a `picture` environment with the `\unitlength` set to 18 mm. Page numbers are set in the headers at the outer sides, and the blobs are attached to these. In this example the section numbers are used to position the blobs, but you can replace this with any numeric value. See figure 4 for the resulting overview page and figure 5 for the code.

**Figure 4** Thumb-index overview page




## 19 Final Remarks

The commands to specify the headers and footers are quite complicated when even- and odd-numbered pages have different layout and we want also to redefine the `plain` page style. Therefore for version 2 of fancyheadings I will create

**Figure 5** Thumb-index code

```
\setlength{\unitlength}{18mm}
\newcommand{\blob}{\rule[-.2\unitlength]{2\unitlength}{.5\unitlength}}

\newcommand\rblob{\thepage
  \begin{picture}(0,0)
    \put(1,-\value{section}){\blob}
  \end{picture}}

\newcommand\lblob{%
  \begin{picture}(0,0)
    \put(-3,-\value{section}){\blob}
  \end{picture}%
  \thepage}

\pagestyle{fancy}
\cfoot{}

\newcounter{line}
\newcommand{\secname}[1]{\addtocounter{line}{1}%
  \put(1,-\value{line}){\blob}
  \put(-7.5,-\value{line}){\Large \arabic{line}}
  \put(-7,-\value{line}){\Large #1}}

\newcommand{\overview}{\thepage
  \begin{picture}(0,0)
    \secname{Introduction}
    \secname{The first year}
    \secname{Specialisation}
...etc...
  \end{picture}}

\begin{document}
\rhead{\overview}\mbox{}\newpage % This produces the overview page
\rhead{} % Front matter may follow here
\clearpage
\rhead[\rightmark]{\rblob} \lhead[\lblob]{\leftmark}
...
```

an easier user interface (the old commands will still work). To give you the opportunity to comment, here is a preview:

The `*head` and `*foot` commands will be replaced by commands `\fancyhead` and `\fancyfoot`. These will have an optional argument that selects which fields to change, e.g. L for the left field, E for even pages, etc. See figure 6. Selectors can be combined so `\fancyhead[LE,RO]{text}` will define the field for both the left header on even pages and the right header on odd pages. There will also be a new command `\fancypagestyle` that allows you to (re)define page styles. One use for this is to redefine the page style `plain` so the `\fancyplain` stuff is no longer necessary. Another use is to allow special page styles to be used with `thispagestyle` or maybe for float pages.

The complicated example in section 6 will then become:

**Figure 6** Selectors

| | |
|---|---|
| E | Even page |
| O | Odd page |
| L | Left field |
| C | Center field |
| R | Right field |
| T | float at Top |
| B | float at Bottom |
| F | Float page |

```
\fancyhead{} % clear all fields
\fancyhead[LE,RO]{\bfseries The performance of new graduates}}
\fancyfoot[LO,CE]{From: K. Grant}
\fancyfoot[RE,CO]{To: Dean A. Smith}
\fancyfoot[RO,LE]{\bfseries \thepage}
\setlength{\headrulewidth}{0.4pt}
\setlength{\footrulewidth}{0.4pt}

\fancypagestyle{plain}{%
\fancyhead{}
\fancyfoot[C]{\bfseries \thepage}}
```

As you can see this is much less complicated.

If you want to comment please use e-mail or regular mail to:

Piet van Oostrum
Dept. of Computer Science
Utrecht University
P.O. Box 80.089
3508 TB Utrecht, The Netherlands
Telephone: +31 30 531806 Telefax: +31 30 513791
Internet: piet@cs.ruu.nl

# Index