Page layout in LaTeX

Piet van Oostrum*
Dept. of Computer Science
Utrecht University

September 6, 1998

Abstract

This article describes how to customize the page layout of your LaTeX documents, i.e how to change page margings and sizes, headers and footers, and the proper placement of figures and tables (collectively called floats) on the page.

Originally this was the documentation of the fancyheadings package. It did contain also other info, e.g. advanced use of marks. It has now been upgraded to include more, e.g. the handling of floats. The fancyheadings documentation has been upgraded to conform to version 2 of this package[1]. For reasons of compatibility with certain operating systems, the name of the package has been changed to fancyhdr.

Although this paper uses LaTeX 2_ε commands, most of the techniques can be used with older LaTeXversions with appropriate changes.

Contents

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

[1]this version is due to be released Real Soon Now

1 Introduction

A page in a LaTeX document is built from various elements as shown in figure 1. The body contains the main text of the document together with the so called floats (tables and figures).

The pages are constructed by LaTeX's output routine, which is quite complicated and should therefore not be modified. Some of the packages described in this paper contains small modifications to the output routine to accomplish things that cannot be done in another way. You should use these packages to get the desired result rather than fiddling with the output routine yourself.

There a a number of things that you must be aware of:

1. The margins on the left are not called `\leftmargin`, but `\evensidemargin` (on even-numbered pages) and `\oddsidemargin` (on odd-numbered pages). In one-sided documents `\oddsidemargin` is used for either. `\leftmargin` is also a valid LaTeX parameter but it has a different use (namely the indentation of lists).

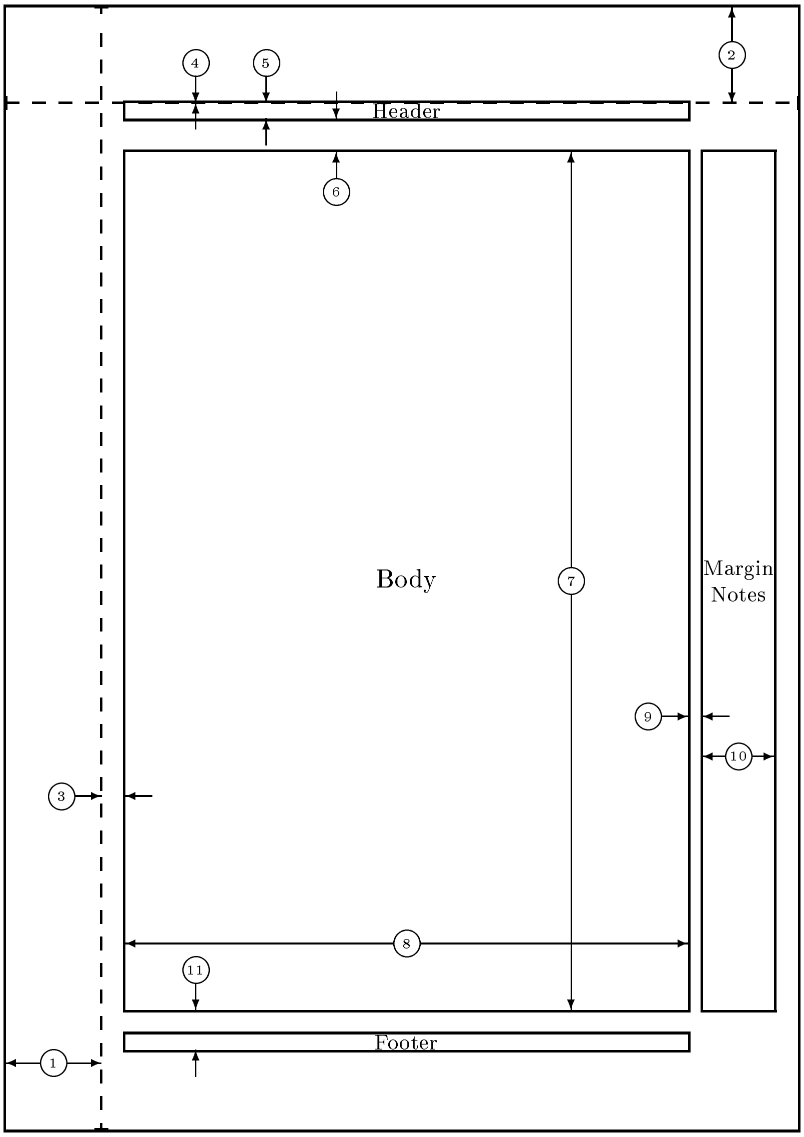
2. Most of the parameters should not be changed in the middle of a document. Some changes might work at a pagebreak. If you want to change the height of a single page, you can use the `\enlargethispage` command.

The margin notes area contains small pieces of information created by the `\marginpar` command. On twosided documents the margin notes appear on the left and right alternatively. The margin notes are not on fixed places with respect to the paper but at approximately the same height as the paragraph in which they appear. Due to the algorithm used to decide the placement of margin notes, in a twosided document unfortunately they may appear on the wrong side if they are close to a page break. If you want to put information on fixed places in the margins you may use the technique described in sections 19 and 20.

The first part of this paper describes how to change the header and footer areas. The last part describes how to get your floats at the desired place.

2 Page headers and footers

The page headers and footers in LaTeX are defined by the `\pagestyle` and `\pagenumbering` commands. `\pagestyle` defines the general contents of the headers and footers (e.g. where the page number will be printed), while `\pagenumbering` defines the format of the page number. LaTeX has four standard pagestyles:



1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 18pt	4	\topmargin = 0pt
5	\headheight = 12pt	6	\headsep = 25pt
7	\textheight = 646pt	8	\textwidth = 424pt
9	\marginparsep = 11pt	10	\marginparwidth = 54pt
11	\footskip = 30pt		\marginparpush = 5pt (not shown)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

Figure 1: Page elements. The values shown are those in effect in the current document, not the defaults.

`empty`	no headers or footers
`plain`	no header, footer contains page number centered
`headings`	no footer, header contains name of chapter/section and/or subsection and page number
`myheadings`	no footer, header contains page number and user supplied information

Although these are useful styles, they are quite limited. Additional page styles can be defined by defining commands of the form `\ps@xxx`. This command is executed when a `\pagestyle{xxx}` is given in the document. The `\ps@xxx` command should define the following commands for the contents of the headers and footers:

`\@oddhead`	header on odd numbered pages in two-sided documents (on all pages in one-sided)
`\@evenhead`	header on even numbered pages in two-sided documents
`\@oddfoot`	footer on odd numbered pages in two-sided documents (on all pages in one-sided)
`\@evenfoot`	footer on even numbered pages in two-sided documents

These are not user commands, but rather "variables" that are used by LaTeX's output routine. As the command names contain the character '`@`', they should be defined in a package file, or otherwise be sandwiched between the commands `\makeatletter` and `\makeatother`.

The `\pagenumbering` command defines the layout of the page number. It has a parameter from the following list:

`arabic`	arabic numerals
`roman`	lower case roman numerals
`Roman`	upper case roman numerals
`alph`	lower case letter
`Alph`	upper case letter

The `\pagenumbering{xxx}` defines the command `\thepage` to be the expansion of the page number in the given notation `xxx`. The pagestyle command then would include `\thepage` in the appropriate place. Additionally the `\pagenumbering` command resets the page number to 1. The `\pagestyle` and `\pagenumbering` apply to the page that is being constructed, so they should be used at a location where it is clear to what page they apply (see section 17).

3 What is fancyhdr

The fancyhdr 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
- different headers and footer on pages with floats

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

4 Simple use of **fancyhdr**

To use this package in a $\LaTeX\,2_{\varepsilon}$ document, place the file `fancyhdr.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[2]:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

We can visualize the page layout we can create with fancyhdr 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.

5 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}` [3]:

```
\lhead{}
\chead{}
\rhead{\bfseries The performance of new graduates}
\lfoot{From: K. Grant}
\cfoot{To: Dean A. Smith}
```

[2] For LaTeX2.09 you should specify `[fancyhdr]` in the `\documentstyle` instead of the `\usepackage` command.

[3] Note that version 1 of fancyheadings used the `\setlength` command to change the `\...rulewidth` parameters.

```
\rfoot{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

(The `\thepage` macro displays the current page number. `\bfseries` is the LaTeX 2_ε'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`.

6 An example of two-sided printing

Some document classes, such as `book.cls`, print two-sided by default: 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:

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

We use the more general commands `\fancyhead` and `\fancyfoot`. These have an additional parameter between square brackets that specifies for which pages and/or parts of the header/footer they apply. The first `\fancyhead` command omits this parameter, and thus applies to all header

E	Even page
O	Odd page
L	Left field
C	Center field
R	Right field
H	Header
F	Footer

Figure 2: Selectors

fields. In general this is only useful to get rid of the defaults or a previous definition, as is done here. The selectors that can be used between the square brackets are given in figure 2. 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. If you don't give an E or O the definition applies to both. Similar for LRC. So the use of `\lhead` in the previous section is just an abbreviation for `\fancyhead[L]`. The selectors may be given as uppercase or lowercase letters.

There is also a more general command `\fancyhf` that you can use to combine the specifications for headers and footers. This allows additional selectors H (header) and F (footer). In fact `\fancyhead` and `\fancyfoot` are just `\fancyhf` with H and F prespecified.

Again, you may use `\thispagestyle{plain}` for a simple page layout for page 1.

7 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 you must redefine the `plain` pagestyle. As we indicated before you could do this by defining the `\ps@plain` command, but fancyhdr gives you an easier way with the `\fancypagestyle` command. This command can be used to redefine existing pagestyles (like `plain`) or to define new ones, e.g. if part of your document is to use a different pagestyle. This command has two parameters: one is the name of the pagestyle to be defined, the second consists of commands that change the headers and/or footers, i.e. `fancyhead` etc. Also allowed are changes to `\headrulewidth` and `\footrulewidth`. As an example, let us redefine the `plain` style for the report in Section 6 by making the page number bold.

```
\fancypagestyle{plain}{%
\fancyhf{} % clear all header and footer fields
\fancyfoot[C]{\bfseries \thepage} % except the center
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}}
```

8 The default layout

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

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

and let fancyhdr 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:

```
\fancyhead[LE,RO]{\slshape \rightmark}
\fancyhead[LO,RE]{\slshape \leftmark}
\fancyfoot[C]{\thepage}
```

The following settings are used for the decorative lines:

`\headrulewidth`	0.4pt
`\footrulewidth`	0 pt

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

9 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 8.

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[4]. You must put the redefinition after the first call of `\pagestyle{fancy}` as this sets up the defaults.

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

Code:	Prints:

`\renewcommand{\chaptermark}[1]{%`
` \markboth{\chaptername`
` \ \thechapter.\ #1}{}}`
	Chapter 2. Do it now

`\renewcommand{\chaptermark}[1]{%`
` \markboth{\MakeUppercase{%`
` \chaptername}\ \thechapter.%`
` \ #1}{}}`
	CHAPTER 2. Do it now

`\renewcommand{\chaptermark}[1]{%`
` \markboth{\MakeUppercase{%`
` \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

Figure 3: Marker variants

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 argument of `\chaptermark`.

Figure 3 shows some variants for "Chapter 2. Do it now" (the last example is appropriate in some non-English languages). The % signs at the end of the lines are to prevent unwanted space. Normally you would continue the lines and remove these % signs[5].

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 `\MakeUppercase` is explicitly given in the definition of `\thebibliography`. Similar for INDEX etc. If you don't want to redefine these commands, you can use the `\nouppercase` command that fancyhdr makes available in the header and footer fields. Note that this may screw other things, like uppercase roman numerals in your headers, so it should be used with care. Essentially this command typesets its

[5] the `\MakeUppercase` command is used in LaTeX 2_ε to generate uppercase text, while in LaTeX 2.09 `\uppercase` is used. The difference is that `\MakeUppercase` also deals with non-ASCII letters. Fancyhdr defines `\MakeUppercase` to be an alias for `\uppercase` if it isn't defined.

argument in an environment where `\MakeUppercase` and `\uppercase` are changed into do-nothing operations.

```
\lhead{\nouppercase{\rightmark}}
\rhead{\nouppercase{\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 fancyhdr.

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 18).

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.

10 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 fancyhdr 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:

```
% 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}
```

Figure 4: Fix marks in twocolumn style

```
\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} % if equal
    {\rightmark--\leftmark}} % if not equal
\fancyhead[LE,RO]{\mymarks}
\fancyhead[LO,RE]{\thepage}
```

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 4, this will be solved[6][7].

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

11 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`) and/or of the footer (`\footskip`), otherwise you may get error messages "Overfull `\vbox` ... has occurred while `\output` is active"[8]. 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{fancyhdr}
\pagestyle{fancy}
\addtolength{\headheight}{\baselineskip}
\renewcommand{\sectionmark}[1]{\markboth{#1}{}}
\renewcommand{\subsectionmark}[1]{\markright{#1}}
\rhead{\leftmark\\\rightmark}
```

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

```
\renewcommand{\headrulewidth}{0.6pt}
```

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

```
\renewcommand{\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}}
```

There is one additional parameter that you can set: `\footruleskip`. It defines the distance between the decorative line in the footer and the top of the footer text line. By default it is set to 30% of the normal line distance. You may want to adjust it if you use unusally large or small fonts in the footer. Change it with `\renewcommand`.

12 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:

[8] If you use `11pt` or `12pt` you will probably also have to do this, because LaTeX's defaults are quite small

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

You must issue these commands *after* the first `\pagestyle{fancy}` command as this will establish the default for `\headwidth`[9].

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

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\fancypagestyle{plain}{%
   \fancyhead{} % get rid of headers
   \renewcommand{\headrulewidth}{0pt} % and the line
}
```

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 $\mathcal{AMS}$-LaTeX book.

Chapter pages have no headers or footers. So we declare

```
\thispagestyle{empty}
```

for every chapter page, and we do not need to redefine plain.

Chapter and section titles appear in the form: 2. DO IT NOW, so we have to redefine `\chaptermark` and `\sectionmark` as follows (see Section 9):

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

```
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\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:

[9] In version 2 of fancyhdr it is usually safe to do it before, but it is still advised to do after.

```
\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:

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
\renewcommand{\chaptermark}[1]%
   {\markboth{\MakeUppercase{\thechapter.\ #1}}{}}
\renewcommand{\sectionmark}[1]%
   {\markright{\MakeUppercase{\thesection.\ #1}}}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\newcommand{\helv}{%
   \fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont}
\fancyhf{}
\fancyhead[LE,RO]{\helv \thepage}
\fancyhead[LO]{\helv \rightmark}
\fancyhead[RE]{\helv \leftmark}
```

13 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 fancyhdr, however, you can specify in each of the header- or footer fields

`\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}}
```

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.

fancyhdr gives you 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.

14 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 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
```

15 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 fancyhdr because they take over the complete page layout. For LaTeX 2ε there is a package `lastpage` available which you can use with fancyhdr 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 2ε you can use the LaTeX2.09 compatible `lastpage209.sty`. which is defined as follows:

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

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[10]:

```
\usepackage{lastpage}
\usepackage{ifthen}
...
\rfoot{\ifthenelse{\isodd{\value{page}} \and \not
    \value{page}=\pageref{LastPage}{please turn over}{}}
```

[10]This requires a reasonably recent version of the `ifthen` package.

16 Chapter or section related page numbers

In technical documentation very often page numbers are used of the form 2-10 where the first number is the chapter number and the second is the pagenumber relative to the chapter. Sometimes section is used rather than chapter. The package chappg can be used to get this format. If you want to change the layout e.g. use a dot rather than a dash, or a section rather than a chapter, you should make a private copy and edit it.

This package redefines `\thepage` as `\arabic{chapter}-\arabic{page}`. Unfortunately this gives numbers rather than letters for appendices. A better definition would be `\thechapter-\arabic{page}` but you can give this definition yourself after the `\usepackage{chappg}` command. What the package also does is reset the page number to 1 at the beginning of each chapter.

There is a fundamental difference between the page numbering of the style "m of n" as described in the previous section and the current one. The m of n style is only used in the page header or footer, but not in the table of contents, index, or references like "*See page* xx". Therefore it does not change the command `\thepage`. The page numbering style "2-10", however should be used in all references to the page number, therefore it must be done by redefining `\thepage`.

17 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 9 and 18. 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 fancyhdr 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 fancyhdr commands make global changes so they will work, as will the `\thispagestyle` command.

It should be noted that although the fancyhdr commands like `\fancyhead` 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 `\fancyfoot[C]{\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.

18 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 5 constitutes a package that gives you two extra marks that can be used in this situation[11]. 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[12]:

```
\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

[11] 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.

[12] Actually there is another way but it requires two LaTeX passes: you can put `\label` commands before and after the text and compare the `\pagerefs`.

```
% 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}
```

Figure 5: Package for extra marks in LaTeX

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 20 for another example of this technique.

```
\lhead{\setlength{\unitlength}{\baselineskip}%
\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. Make sure you 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 set this in one of the fancyhdr 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`.

19 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 fancyhdr. 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 `graphics` or `graphicx` package[13]. To put the movie in the righthandside bottom corner the following will work:

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

Notice that the `\unitlength` parameter should be set locally in the fancyhdr field in order to avoid unwanted interference with its value in the text.

20 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

[13] If you use an older version of LaTeX you could use the `epsf` or `epsfig` package.

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 6 for the resulting overview page and figure 7 for the code.

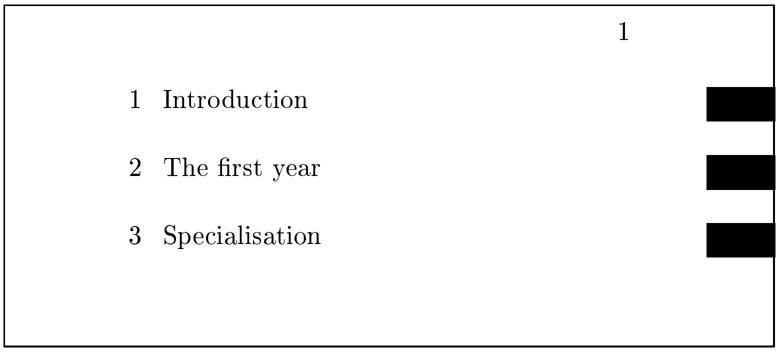


Figure 6: Thumb-index overview page

21 Float placement

Floats are page elements that float with respect to the rest of the document. Standard floats are tables and figures, but with the float package you can easily make new ones, like algorithms. Most of the time floats work satisfactory, but sometimes LaTeX seems too stubborn to do what you want. This section describes how you can influence LaTeX so that it will do most of the time what you want. There might, however be some pathological cases where it is impossible to convince LaTeX to do things your way. In the following we will use figures as an example but everything applies to other floats as well.

The most encountered problems with floats are:

1. You want a float at a certain position in the text, but LaTeX moves it, usually to the next page.
2. From a certain point, LaTeX moves all your floats to the end of the document or the end of a chapter.
3. LaTeX complains about "Too many floats".

In the first two cases you must first check if you have given the correct "placement" parameter to you float, e.g. `\begin{figure}[htp]` specifies that your figure may be placed either: Here (i.e. in the text position where the command is given), on the Top of a page (which may be the page where you put the command), or on a separate Page of floats. You could also have specified "b" for Bottom of the page. The order of the letters is insignificant, you cannot force LaTeX to try Bottom first and then Top by specifying `[bt]`.

If LaTeX doesn't put the float at the place where you expected it, it is usually caused by the following:

```
\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}
\fancyhead[R]{\overview}\mbox{}\newpage % This produces the overview page
\fancyhead[R]{} % Front matter may follow here
\clearpage
\fancyhead[RE]{\rightmark}
\fancyhead[RO]{\rblob}
\fancyhead[LE]{\lblob}
\fancyhead[LO]{{\leftmark}
...
```

Figure 7: Thumb-index code

1. The float didn't fit on the page. In this case it has to move to the next page or even further. If you didn't specify either `[t]` or `[b]` in the position parameter, LaTeX must save it until it has enough for a page of floats. So don't specify only `[h]`. If you want to give LaTeX a chance to put the float on a page of floats, you must also specify "p".

2. The placement would violate the constraints imposed by LaTeX's float placement parameters. This is one of the most occurring causes and it can easily be corrected by changing the parameters. Here is a list of them:

Counters – change with `\setcounter`		
`topnumber`	max. number of floats at top of page	2
`bottomnumber`	max. number of floats at bottom of page	1
`totalnumber`	max. number of floats on a page	3
Other – change with `\renewcommand`		
`\topfraction`	max fraction of page for floats at top	0.7
`\bottomfraction`	max fraction of page for floats at bottom	0.3
`\textfraction`	min fraction of page for text	0.2
`\floatpagefraction`	min fraction of floatpage that should have floats	0.5

There are also some others for double column floats in two-column documents.

The values in the righthand column are the defaults for the standard LaTeX classes. Other classes could use different defaults. As you see with the default values a float will not be put in the bottom of a page if its height is more than 30% of the page height. So if you specify [hb] for a float which is taller it has to move to a float page. But if it is less than 50% of the page height it will have to wait until some more floats are given before a float page can be filled to satisfy the `\floatpagefraction` parameter. If you have this kind of behaviour you can easily adapt the parameters, e.g. with:

```
\renewcommand{\textfraction}{0.05}
\renewcommand{\topfraction}{0.95}
\renewcommand{\bottomfraction}{0.95}
\renewcommand{\floatpagefraction}{0.35}
\setcounter{totalnumber}{5}
```

You may want to be careful not to make `\floatpagefraction` too small, otherwise you may get too many small floatpages.

You can force LaTeX to ignore most of the parameters for one specific float occurrence by including an exclamation mark (!) in the placement parameters, e.g.

```
\begin{figure}[!htb]
```

Floats which contain a "`t`" in the position parameter could be placed before the place where they are referenced (but on the same page). This is normal behaviour for LaTeX but some people just don't like it. There are a number of ways to prevent this:

1. Of course deleting the "`t`" will help, but in general this is undesirable, as you may want the float to be placed at the top of the next page.

2. use the `flafter` package which causes floats never to be placed "backwards".

3. use the command `\suppressfloats[t]`[14]. This command will cause floats for the top position *on this page* to be moved to the next page. This can also be done with `[b]` or without parameter for all floats on this page.

[14] This command and the "!" placement parameter are not defined in LaTeX2.09.

If in spite of all your attempts LaTeX still moves your floats to the end of the document or the end of a chapter, you can insert a `\clearpage` command. This will start a new page and insert all pending floats before continueing. If it is undesirable to have a pagebreak you can use the afterpage package and the following command:

```
\afterpage{clearpage}
```

This will wait until the current page is finished and then flush all outstanding floats. In some pathological circumstances `afterpage` may give strange results, however.

Finally, if you want a float only at the place where you define it, without LaTeX moving it whatsoever, you can use the float package and give the command:

```
\restylefloat{figure}
```

in the preamble. Now you will be able to specify `[H]` as the position parameter, which will mean "HERE and only HERE". This may cause an unwanted page break however[15]. If you want to avoid the unwanted pagebreak, i.e. let LaTeX move the float only if it doesn't fit on the page, the use the afterpage package with:

```
\afterpage{\clearpage \begin{figure}[H] ... \end{figure}}
```

Complaints from LaTeX about "Too many floats" are usually caused by one of the above problems: floats not being able to be placed and LaTeX collecting too many of them. The solutions given above, especially those with `\clearpage` in them will usually help. In some cases there really are too many floats, as LaTeX has a limited number of "boxes" to store the floats. The package morefloats can be used to increase this number. If you need still more then you must edit a private copy of this file, but even then there will be some limit that you cannot pass. Then your only resort will be to change your document.

22 Multipage Floats

LaTeX's floats cannot be split across pages. Sometimes, however, you want to have a table or figure that doesn't fit on one page. The easiest way is to split these into multiple table or figure environments, but this has a number of undesirable effects:

- Where do you split it? This is in general more difficult for tables than for figures.
- How do you keep them together?
- You don't want more than one entry in the list of figures/tables.

Although these problems are not fully solvable in all cases, here are a couple of suggestions:

22.1 Tables

For tables longer than a page you can use the `longtable` package. This package defines a `longtable` environment that is a kind of amalgamation of `table` and `tabular`. It has approximately the same syntax as the `tabular` environment, but it adds some features of `table`, like captions. Longtables will be automatically split when they don't fit on the page. And they will be entered in the list of tables when a caption is given. They will not float, however, and cannot be used inside a float environment. This could mean that another `table` environment, which was

[15] There used to be a `here.sty` with the same effect, but this is incompatible with LaTeX 2_ε

defined before the `longtable`, will float past it, and therefore the numbers may get out of order. Another problem could be that the `longtable` starts rather far down the page, which isn't a pleasant sight. If you want the `longtable` to start at the top of the page, the best thing to do is to include it in an `\afterpage` command (using the `afterpage` package). As a `longtable` is by definition large, it is best to put it in a separate file, and `\input` it in the `\afterpage` command:

```
\afterpage{\input{mytable}}
```

```
\afterpage{\clearpage\input{mytable}}
```

The last form has the additional advantage that most of the outstanding floats will be printed first.

22.2 Figures

There isn't an equivalent `longfigure` solution, so for figures you will usually have to split it yourself. In general this is less of a problem. However, the problem you get now is how to keep them together, i.e. how to get the parts on subsequent pages, and how to get a single entry in the list of figures.

You will have to split the figure into pieces and put each part in a separate `figure` environment. To keep them together it is best to use only the `[p]` placement, so that they will be put on floatpages. As they are bigger than a page this is appropriate. The first part would then get a `\caption`, the subsequent parts would be used without a caption. If you want to add a caption-like text, enter it as normal text rather than a `\caption`, so that it will not be entered in the list of figures. It may also be desirable to issue a `\clearpage` first, just like we did for the `longtable`, and to encapsulate this in the `\afterpage` command. E.g.

```
\afterpage{\clearpage\input{myfigure}}
```

where `myfigure.tex` contains:

```
\begin{figure}[p]
\includegraphics{myfig1.eps}
\caption{This is a multipage figure}
\label{fig:xxx}
\end{figure}
\begin{figure}[p]
\includegraphics{myfig2.eps}
\begin{center}
Figure~\ref{fig:xxx} (continued)
\end{center}
\end{figure}
```

You have to make sure that the last part is big enough, otherwise LaTeX could decide to postpone it until it has collected some more floats. This can be done either by making the figure big enough (e.g. by adding some `\vspace`), or by tweaking the `\floatpagefraction` parameter.

If you want your multipage figure to start at a lefthand-side (even-numbered) page you can use a test in the `\afterpage` command (using the `ifthen` package):

```
\afterpage{\clearpage
\ifthenelse{\isodd{\value{page}}}{\afterpage{\input{myfigure}}} % odd page
    {\input{myfigure}}}} % even page
```

If there are too many floats on the skipped page, this may still fail to start your multipage figure on an even page, however.

23 Contact information

Piet van Oostrum
Dept. of Computer Science
Utrecht University
P.O. Box 80.089
3508 TB Utrecht, The Netherlands
Telephone: +31 30 2531806 Telefax: +31 30 2513791
E-mail: piet@cs.ruu.nl
WWW: http://www.cs.ruu.nl/people/piet

Index