Changing the layout with LaTeX

EDV-Zentrum der TU Wien, Abt. Digitalrechenanlage
Hubert Partl
1988-10-04
English translation by
*Axel Kielhorn**
1999-03-20
H27.0 — Version 1

Abstract

This document describes the capabilities of the `refart` and `refrep` classes for LaTeX 2ε. These classes do not work with LaTeX 2.09. They contain some improvements over the original `refman` style which may result in different output and minor incompatibilities, but make refman work with paper sizes other than ISO A4, which I consider an improvement.

This manual is an addition to chapter 5 ("Designing It Yourself") of the LaTeX-manual by Leslie Lamport. It was originally written in 1988 by Hubert Partl and updated by me during the development of `refman` 2.0. The translation into English was done in summer 1998, almost 10 years after the initial release in German.

Contents

1 Introduction **3**

1.1 Author, Layout Designer and Compositor 3

1.2 Changing the Layout 3

2 The Art of Layout-Design **3**

2.1 Common Rules 3

2.2 Special note for technical descriptions 4

3 How to change a LaTeX layout? **5**

3.1 Advantages and Disadvantages of the text processing system LaTeX 5

3.2 Input files and class files 6

3.3 Class files and Packages 6

3.4 Changing the layout, step by step 7

* a.kielhorn@web.de

3.5 A simple example (Equation numbers) 8
3.6 A more complex example (Reference Manual) 9

Appendix **12**

A The page structure in LaTeX **12**

B Description of the refman-class family **15**
B.1 Invocation 15
B.2 Options 15
B.3 Layout changes 15
B.4 Footnotes 16
B.5 Additional commands 17

1 Introduction

1.1 Author, Layout Designer and Compositor

Every piece of writing – from a short handwritten note to a printed book – consists of two components: **contents** and **design.** In most cases the design is only a means to allow the reader to understand the contents more easily.

Author: The **contents** will be written by the author of the document. She usually is supported by the publisher, who issues guidelines about how the document should be written and conducts proofreading.

When using LaTeX the contents appears in the main part of the document – between `\begin{document}` and `\end{document}` – with some additional LaTeX-commands that describe the meaning of the different parts.

Designer: The **design** of a printed document is usually defined by the publisher who employs designers for this purpose.

The layout is defined in so-called class files. When writing a document you have to specify the class you want to use with the `\documentclass` command.

The document classes distributed with LaTeX have been designed by American layout designers according to the conventions used in the USA. The "Koma-Script" classes have been adapted to European taste and contain many small improvements.

Compositor: The compositor takes the contents from the author and formats it according to the layout the designer specified.

When using LaTeX, this job is done by the TeX program and the printing is done by a device driver.

1.2 Changing the Layout

If you want to use a layout different from the ones distributed with LaTeX, you have to take the following steps:

1. The layout has to be defined. This is usually a job for a professional designer.

→ Chapter 2

   You can find detailed information about how change the design in chapter 2.

2. The layout has to be programmed in LaTeX. This can either result in a "package" that changes the behaviour of a standard class, or in the definition of a new "class". `Refman` used to be a package that changed the definition of the `article` class but is now a class of its own. When writing a new class, you will have more work, but as a result more freedom to change things.

→ Chapter 3

   Chapter 3 contains more information about the new layout.

2 The Art of Layout-Design

2.1 Common Rules

There are almost no common rules because every kind of document has different requirements and needs a specific layout. This layout should

consider *who* will read the document, and *how*.

An important criteria is if the reader will read the document from start to end like a detective novel (linear reading) or if he wants to find certain information as in a telephone directory or a reference manual.

In addition to that, the layout has to consider certain conventions, like the habits of the reader or "Corporate Design" rules that distinguish publications from different publishers. [1]

! → The main purpose of a layout is to make sure the reader finds the information he wants and is able to read and understand it easily. The structure, readability, and consistency of a document is more important than it's "beauty".[2]

The following "rules of thumb" will be valid for most applications:

Line spacing : The spacing between two lines should be larger than the spacing between two words to guide the eyes of the reader.

Line length : The length of a line – or when using multicolumn layout of a column – should be about 60 characters. When lines get longer they are more difficult to read and it is easier to go to the wrong line after finishing the current one. Increasing the linespacing may help a little. When lines get to short it is difficult to set them justified, and you will get lots of hyphenated words.

Page layout : Normal text pages should look the same throughout the document. Figures, tables and special pages like the index need not appear in the same layout but should take as much space as needed.

Margin notes : Margin notes are often more suitable than footnotes because they appear right next to the text they refer to. Special margin notes are the "attention sign" or the "dangerous bend" that guide the user to important parts of the text.

Headings and Footings : Headings and footings should make it easier for the reader to orient himself in the document. If you expect readers to copy single pages from the document they should contain information about the paper as a whole, just in case you need more information or want to cite the whole paper.

If you expect the document to change often (like software manuals), each page should contain a version information or at least a date.

2.2 Special note for technical descriptions

Let us compare three different layouts and check if they are usable for technical descriptions, user's guides and reference manuals. [3]

Plain TEX : The standard format use by plain TEX has the great disadvantage that the lines are much to long, which reduces the readability.

Standard-LATEX : The format used by LATEXs standard classes isn't ideal. The line length is correct, but that leaves us with a wide unused margin. The font used for section headings is too large. The Koma-Script classes improve this and offer many ways to configure the fonts.

[1] Compare the layout of different daily papers or magazines like "page" or "invers".
[2] This is not always true for adverts that usually contain no information at all and, picture magazines where the beauty of the picture *is* the contents.
[3] This hint came from Paul Stiff, who teaches layout design at the University of Reading.

This is a good design for a paper one usually reads from start to end.

Reference Manual Style : A new design that appeared some years ago and is used in recent reference manuals, is much more suited for our purpose. [4]

- The text is printed in rather short lines in the right part of the page. This part is used for continuous reading.
- The wide left margin is used for headings and margin notes. Since you now have a wide margin it is easier to use long margin notes to supply additional information and to lead the reader to important parts of the documents. Please note that the margin is always on the left side thus two-sided printing does not look symmetrical. This is done on purpose, because the reader will always start reading at the left side, and with this layout section headers really "stand out". In a symmetrical layout, half the headers would be buried in the text.
- Figures and tables are either inside the text column, inside the margin or, if necessary, fill the whole page.

→ Section 3.6 Section 3.6 describes how to implement such a layout in LaTeX.

3 How to change a LaTeX layout?

3.1 Advantages and Disadvantages of the text processing system LaTeX

Advantages: The big advantage of LaTeX is, that it implements a "generic" or "logical" design. This means that the author has to specify the *meaning* of special parts of the text like: headings, citations, lists, literature references, and so on. These logical definitions will be processed by the system and printed in the "correct" way. The meaning of *correct* is defined in the document class and additional packages.

The opposite of this is the "visual" design that most text processors use. Here the author has to know the correct way to set certain parts of the text and take care of the correct printing.

The logical design makes it easier for the author to write consistent documents (i. e., same font and fontsize for section headings of the same level, same layout for lists and enumerations, ...).

Disadvantages: The main disadvantage of LaTeX is that the author has only limited means to change the layout and that she has only four classes to choose from. This has changed a great deal with the appearance of the Script classes for LaTeX 2.09 in 1992, which in turn have been replaced by the even more improved Koma-Script classes for LaTeX 2_ε.

Another disadvantage is that LaTeX seems to be tied to the "Computer modern" font family. This is simply not true, at least not when you have a PostScript printer. Setting up different fonts isn't an easy task, but once they are installed they are as easy to use as the standard fonts.

But LaTeX is much more powerful and flexible: you can define an arbitrary design by changing the definitions in the class files or overwrite them by packages. This is easier than you may think and happens more often than you think. Many universities and publishers have their own LaTeX class but do not distribute it to the world.[5]

[4] The "PostScript Reference Manual" is one document that uses such a design.

[5] You can find a generic `elsevier.cls` on CTAN which you can use to prepare arti-

3.2 Input files and class files

According to the principle of separation of content and design, there are two kind of files:

- The content and the logical structure of a document are defined in the LaTeX input file.
- The design (layout) is defined in the class files and packages.

Which class and packages files a document will use is defined at the beginning of the input file. The `\documentclass` command selects the class and the `\usepackage` command specifies additional packages.

To generate a paper you need at least two files, the input file and a class file.

These two files represent the work of the author and designer as explained in the introduction. Even if the author and the designer are the same person, this has some advantages:

- Similar documents (that appear in a series) have the same layout because the layout is defined in a file of its own and not part of the document.
- You can print the same contents without much work in different layouts, e. g.. as an article for a magazine and as a chapter for a dissertation.

3.3 Class files and Packages

LaTeX supports a hierarchy of layout definitions.

- The first file processed is the class file that is specified inside the curly braces of the `\documentclass` command. This defines the kind of document you want to write.
- The optional argument of the `\documentclass` command inside the square brackets defines class options which select variants of the basic layout,such as different font sizes.
- The last step is reading the packages specified by the `\usepackage` command. This command again takes options to select the layout.
- You can change layout parameters in the input file, but this is discouraged because it violates the principle of separation of content and design.

There are some important differences between class and package files and "normal" input files:

- Class and package files should only contain definitions. They must not output text.
- The "at"-sign `@` is treated as a letter and therefore may appear in command names. Most internal commands of LaTeX contain an `@` to prevent the author from using them accidentally.
- The extension of the file is `.cls`, `.clo` or `.sty` instead of `.tex`.

cle before submission. This will be replaced by the magazine-specific class at the publisher.

3.4 Changing the layout, step by step

It is usually easier to change existing class files instead of writing a new one from scratch. In many cases it is even sufficient so replace some definitions and put them into a package instead of creating a new class.

Please note that you are *not* allowed to change the standard classes distributed with LaTeX. You *have* to change the name when you want to make changes. That is another reason to put small changes in packages.

3.4.1 Defining the differences between the desired and the available layout

The first step is to define the difference between the layout you have and the layout you want.

3.4.2 Finding the original definition

The next step is to find out where the original layout is defined. It is best to search the files in the following order:[6]

1. the LaTeX manual by Leslie Lamport,
2. the LaTeX documentations files `*.dtx` for the classes or packages
3. the LaTeX documentations files `*.dtx` for the kernel,
4. the TeXbook by Donald E. Knuth.

The files are usually documented quite well so you should be able to change things even if you don't understand everything.

3.4.3 Writing a new package file

The third step is to create a new package. You choose an appropriate name for the package (like `mysty`) and create a filename by adding the extension `.sty`.

This file will only contain the definitions you want to change or the new commands you want to define.

If you want to change definitions or certain parameters, the best way is to copy them from the original file and modify them according to your liking.

Defining new commands is easier when you find similar commands in the original files which you can change.

It is always a good idea to include the reason you wrote the package, the changes it makes and the new commands it defines in the file. You should include the date of the last change and the LaTeX version it works with, just in case some internal LaTeX commands you use will change.

When writing larger packages, it is an even better idea to use the `docstrip` program which is used to document the LaTeX 2_ε files. Thus you have your code and documentation in one file and it's easier to keep them from going out of sync.

[6] This hint came from Sue Brooks, who held a workshop for "LaTeX-Hacker" at the 1988 TeX-conference in Exeter.

3.4.4 Using the new package

To use the new package, you call it with the `\usepackage` command. This command executes the code of your package and changes the layout as desired.

Example:

```
\documentclass[11pt,twoside,a4paper]{article}
\usepackage{mysty} %<- This calls the package "mysty"
```

You shouldn't need to change anything else in your input file, unless you defined new commands or environments that are not available in standard LaTeX.

! → When you copy your input file to a different computer you have to include your new packages as well. Otherwise the document can't be processed.

3.5 A simple example (Equation numbers)

Let's assume that you want to write an article where the equations are numbered separately in every section. In the LaTeX manual you find a notice that the report class does something similar for every chapter.

Looking into the file `report.cls` you will find the following commands that deal with equation numbers:

```
\@addtoreset{equation}{chapter}
\def\theequation{\thechapter .\arabic{equation}}
% or in LaTeX2e since 1995/06/01:
\renewcommand\theequation{
             \thechapter.\@arabic\c@equation
                          }
```

You don't necessarily need to understand these two commands in detail.

Now you create an new file with the name `eqpersec.sty`[7] and copy the commands above into that file. After that you replace every occurrence of `chapter` with `section` and add some comments.

```
% This is equation_per_section.sty
% Short name: eqpersec.sty
% Original file by Hubert Partl 1988
% Modified by Axel Kielhorn 1996/01/01
% to support LaTeX 1995/06/01 and later
%
% reset the equation counter at the start
% of a new section
%
\@addtoreset{equation}{section}
% Equationnumber = sectionnummer.equationnummer
% Use only one of the below
% depending on you LaTeX version
%
%\def\theequation{\thesection .\arabic{equation}}
% or in more recent versions of LaTeX
\renewcommand\theequation{
```

[7] Depending on the computer you are using the name may be different like `EQPERSEC_STY` on a CYBER running NOS/VE. But note that you must not use spaces in the filename.

```
\thesection.\@arabic\c@equation
}
```

Whenever you use a `\usepacakge{eqpersec}` command as in

```
\documentclass[11pt]{article}
\usepackage{eqpersec}
```

you will get equations numbered according to your conventions.

3.6 A more complex example (Reference Manual)

We want to create a layout similar to the one used in the *PostScript Reference Manual*, with a wide left margin for headings and margin notes and a small margin at the right and bottom.

3.6.1 Page layout

To define the new layout we use the commands described in the LaTeX manual. For full details see the file `refman.dtx`.

Horizontal: First we define two new names for length that we will use often:

`\fullwidth` is the width of the whole page minus a margin of 1 inch on every side.

$$\texttt{fullwidth} = \texttt{paperwidth} - 2 \text{ inch}$$

From this the width of the text is calculated.

$$\texttt{textwidth} = \texttt{fullwidth} \times \texttt{textfraction}$$

`\leftmarginwidth` is the width of the left margin that will be used for headings and margin notes.

$$\texttt{leftmarginwidth} = \texttt{fullwidth} - \texttt{textwidth}$$

This is a little more difficult in reality because the lengths have to be rounded to full points and a possible two column layout – as used in the index – must be taken into consideration.

Vertical: The vertical layout is a little more difficult because you have to deal with the page header and footer.

$$\texttt{textheight} = \texttt{paperheight} - 2.5 \text{ inch}$$

The result of this calculation is rounded to full lines. Depending on the page style – headings or footings – it is shifted up or down by one line.

3.6.2 Section headings

The headings have to be modified to make them extend into the left margin.

In file `classes.dtx` we find the `\@startsection` command that defines the layout of the headings. Only parameters 4 to 6 are relevant for us: parameter 4 is the space above and parameter 5 the space below the section. The 6th parameter does the actual formatting.

This is the original definition:

```
\newcommand\section{\@startsection
             {section}{1}{\z@}%
             {-3.5ex plus -1ex minus -.2ex}%
             {2.3ex plus .2ex}%
             {\normalfont\Large\bfseries}}
```

The commands for sub- and subsubsections are similar. Note that the measures are all in `ex`, thus depending on the font size used.

We define a new command `\secshape` to format the headings. This command uses the whole width of the page for the heading. To discourage hyphenation of the heading we give it a high penalty. This still allows hyphenation when absolutely necessary.

```
\newcommand\secshape{%
               \leftskip=-\leftmarginwidth%
               \rightskip=\@flushglue%
               \hyphenpenalty=2000}
```

This command is inserted into the 6th parameter of `\@startsection`.

Since the headings now extend into the left margin, we can use a smaller font and reduce the space between the text and the heading. The new definition looks like the following:

```
\newcommand\section{\@startsection
             {section}{1}{\z@}%
             {-2ex plus -1ex minus -.2ex}%
             {0.5ex plus .2ex}%
             {\secshape\normalfont\large\bfseries}}
```

3.6.3 Setting the margin notes

The margin notes should always appear on the left side of the text. The normal layout puts them into the outer margin in twoside layout.

The file `latex.dtx` contains the definition of the `\@addmarginpar` command which is responsible for the margin notes. We don't have to understand the whole definition; the important part is the internal variable `\@tempcnta` that is either `\@ne` (1) when the note should appear on the right side of the text or `\m@ne` (-1) when it should appear on the left side.

This is done by the following lines:

```
\@tempcnta\@ne
\if@twocolumn
    \if@firstcolumn \@tempcnta\m@ne \fi
\else
  \if@mparswitch
     \ifodd\c@page \else\@tempcnta\m@ne \fi
  \fi
  \if@reversemargin \@tempcnta -\@tempcnta \fi
\fi
```

which we simply replace by:

```
\@tempcnta\m@ne
```

The remaining lines that handle the setting of the margin note depending on the parameter `\@tempcnta` are left unchanged.

3.6.4 Extensions

→ Appendix B

The definitions described above are sufficient for simple applications but in practical use one may want some additional commands. You will find the description for the whole new class in the appendix B.

Appendix

A The page structure in LaTeX

→ Fig. 1

This appendix describes how the actual page is build from its components and how they are influenced by TeX's parameters. (Figure 1 has been created by Nelson Beebe at the University of Utah.)

text area : The normal text area ("Body") contains the running text including footnotes, tables and figures. The headings, footer and margin notes do *not* belong to the text area.

The text area has the width `\textwidth` and the height `\textheight`.

In a two column layout the text area is split into two columns, with the width `\columnwidth` each and a space of `\columnsep` between them. Thus the `\columnwidth` is a little bit smaller than half the `\textwidth`.

`\textwidth` and `\columnwidth` should be a multiple of the width of one character in the `tt` font.

`\textheight` should be multiple of the line height `\baselineskip`, increased by the constant value of `\topskip`.

Indentations inside the text area are defined with `\leftskip` and `\rightskip`. These parameters should not be changed explicitly by the user but rather implicitly through environments.

left margin : The left margin is either `\odd-` or `\evensidemargin` plus 1 inch. Both parameters have the same value, unless the `twoside` option is given.

top margin : The top margin is the sum of `\topmargin`, `\headheight` and `\headsep` plus 1 inch.

right margin : The right margin is the paper width minus the left margin and the text area.

bottom margin : The bottom margin is the paper height minus the top margin and the text area.

heading : The heading is inside the top margin with a space of `\headsep` between the lower border of the header and the upper border of the text area. Above the header is a free space of `\topmargin` increased by 1 inch.

footing : The footer is inside the bottom margin with a space of `\footskip` between the lower border of the text area and the lower border of the footer.

margin notes : Margin notes are inside the left or right margin. They have a width of `\marginparwidth` and a space of `\marginparsep` between the margin note and the text area. The vertical space between two margin notes is `\marginparpush`.

The `paperheight` consists of the following elements (from top to bottom):

1 inch
`\topmargin`
`\headheight`
`\headsep`
`\textheight`
`\footskip`
remaining page.

On pages with margin notes in the right margin the `paperwidth` consists of the following elements:

1 inch
`\oddsidemargin` or `\evensidemargin`
`\textwidth`
`\marginparsep`
`\marginparwidth`
remaining page

With the option `twoside` the left pages change to

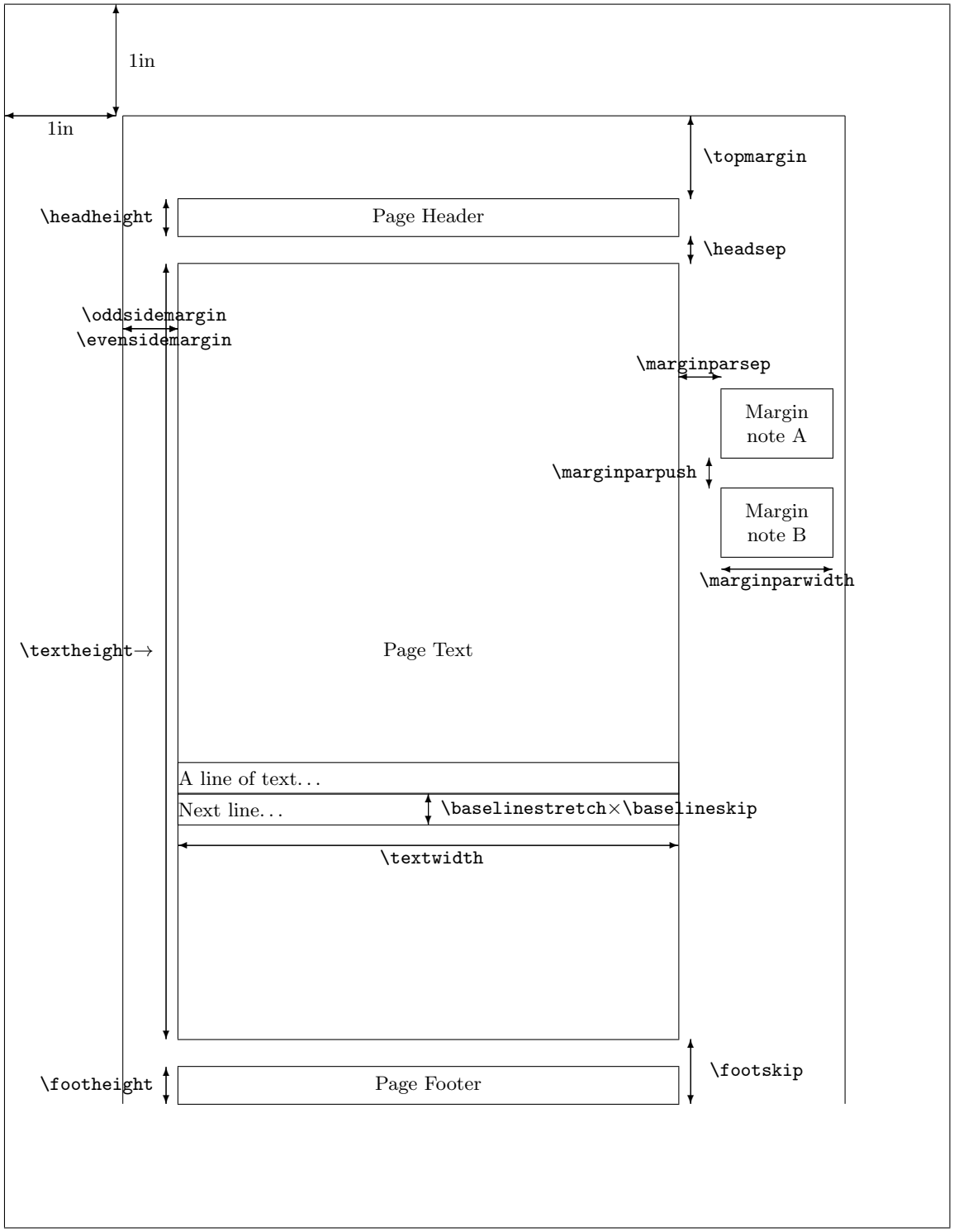
1 inch
`\evensidemargin`
`\textwidth`
remaining page

Comments: The parameters `\topmargin`, `\oddsidemargin`, and `\evensidemargin` may be negative. In this case, the margin will be smaller than 1 inch. The same is true for `\leftskip` and `\rightskip` which leads to text that is wider than the text area.

Extensive treatment and figures to this subject may be found in the TUGboat Vol.9, No.1 (April 1988).

The parameter `\footheight` is no longer defined in LaTeX 2_ε since no-one used it.

Figure 1: Page layout



B Description of the refman-class family

The `refman.sty` was defined at the EDV-Zentrum (computing center) of the TU[8] Wien. This layout is suitable for reference manuals, technical descriptions and similar applications. It is based on the ideas shown in previous sections: The layout has a wide left margin for headings and margin notes and smaller margins on the right side, the top and the bottom.

In 1994 this layout was re-implemented as a class for the new LaTeX 2_ε. This made it possible to include some minor improvements, such as the support of different paper sizes. The `refman.sty` was split into two classes `refrep`, similar to `report` and `refart`, similar to `article`. These classes differ in the layout of the header and footer. The `refart` does not support the `\chapter` command.

The current version of both classes is described in this document. It serves as an example for the layout.

B.1 Invocation

The LaTeX local guide (if available) shows if this class is available at your TeX installation or where to install it. To use the `refart` class, simply call it with the `\documentclass` command:

```
\documentclass[11pt,a4paper]{refart}
\usepackage{german} % other packages you may want
```

B.2 Options

The `refart` class replaces `article` and `refrep` replaces `report`. They support all options of these classes except for the `twocolumn` option.

It supports the additional option `square` which makes the `\textheight` equal to the `\textwidth`.

Neither `refart` nor `refrep` support two column layout, thus the commands `\twocolumn` and `\onecolumn` must not be used.

The index will be set in two column format and you can't change it with the means of this class.

B.3 Layout changes

B.3.1 Page design

Horizontal: In this design the usable area for text (`\fullwidth`) is calculated as the paper width minus 2 `\papermarginwidth`. The default value for `\papermarginwidth` is 1 Inch.

The option `smallborder` reduces `\papermarginwidth` to 0.25 Inch. This is more suitable for documents viewed on screen, especially when combined with the `a5paper` and `landscape` options.

Only a fraction of this width is used for the running text (`\textwidth`), the remaining part forms a wide left margin (`\leftmarginwidth`) which is used for headings and margin notes. The `\textwidth` is 70 % of the `\fullwidth` by default, but this can be changed with

[8] Technical University

the `\settextfraction` command which accepts arguments between 0 and 1.

Vertical: The text height is calculated as the paper height minus 2 `\papermarginwidth`. The `topmargin` is modified by some pagestyles. (see B.4.3).

The pages are always set with a ragged bottom.

B.3.2 Section headings

The headings for `\section`, `\subsection`, and `\subsubsection` extend into the left margin, thus using the full width of the page. They are not justified and hyphenation is discouraged. A small space is kept free above and below the heading. Headings for `\section` and `\subsection` are set in a bold font.

The `refrep` class defines a different layout for the `\chapter` command: It always starts a new page and prints the chapter headings in a large bold font with a thick line above and below. This heading uses the full width of the page.

A similar heading is created by the `\part` commands which is available in both classes. It uses a roman part number instead of the arabic section number.

The `\maketitle` commands sets the title of the document in the same layout when no special title page is requested. (This is the default for `refart`. To suppress the title page in a `refrep` document, you can use the `notitlepage` option.) The name of the author and the date is printed in italic flush right below the document title.

B.3.3 Paragraphs

Paragraphs are separated by a vertical space (`\parskip`) of half a line (`0.5\baselineskip`) plus a stretchable length of 2 pt. Paragraphs are not indented.

The vertical spacing inside, above and below a list environment is the same as in the running text.

B.4 Footnotes

The footnote layout consists of a small margin (1em) which contains the footnote symbol. A small space is set between the symbol and the footnote text. The paragraphs of the footnote are not indented. There is currently no space between two footnotes, I'm not sure it this will stay this way. The footnote symbol is set as a superscript. This may change in later versions. I'm relying on user feedback to finally solve this.

B.4.1 Description environment

The `description` environment will use the whole left margin for the description label.

→ Section 3 You will find examples in the section 3.

B.4.2 Positioning of margin notes

Margin notes (`\marginpar`) are always put into the left margin. They use the whole width of the margin.

The minimum space between two margin notes is set to 0 to prevent them from being shifted around when many margin notes are used.

B.4.3 Headers and Footers

The page style `plain` puts the page number into the footer in the right corner. When the option `twoside` is active, the page number of left pages is put into the left corner.

The pagestyles `headings` and `myheadings` create a header which spans the whole width of the page. The headings contain the running head (`\section` and `\subsection` in `refart` and `\chapter` and `\section` in `refrep`) when `headings` is used or a fixed text that can defined with the `\markboth` command when `myheadings` is used. The heading will be set in a slanted font and separated from the body by a thin line.

In addition to the standard classes, `refman` supports a style for footers, which is used in this documentation. The information is exactly the same as in the headings but now printed in the footer with a thin line above.

To use a user-defined string you can say:

```
\pagestyle{myfootings} % or myheadings
\markboth{left title}{right title}
```

The `heading` and `myheading` commands increase the top margin by one line while the `footings` and `myfootings` commands decrease the top margin by one line. The page styles `empty` and `plain` leave the top margin unchanged. You should not combine headings and footings in one document.

User feedback has shown that it is not a good idea to combine `plain` and `(my)heading` either. Therefore I changed the layout of the `\chapter` page to `empty`. Maybe it is necessary to define a `hplain` and `fplain` pagestyle or to define some magic to use the correct definition of `plain`. Feedback is welcome.

B.5 Additional commands

B.5.1 Marginlabel

The command `\marginlabel{xxx}` prints the text `xxx` right justified into the left margin. Please note that a `\marginpar` will print it left justified.

Example: The word "Example" in the left margin is printed with the command `\marginlabel{Example:}`

B.5.2 Attention

! → The command `\attention` puts an exclamation mark with an arrow pointing to the text into the left margin. This is an example for `\attention`.

`:-)` Since version 2.0c you can change the symbol used for the `\attention` command using a `\renewcommand{\attentionsymbol}{\texttt\{:-)\}}` command. To get the default back use `\renewcommand{\attentionsymbol} {\large \bfseries ! $\rightarrow$}`

`:-(` Since version 2.0c `\attention` takes an optional argument to define the symbol used in the margin. Thus you can change the symbol once, without having to restore it later. Do not abuse this feature, it is primarily

meant as an support for the `manfnt` package which enables you to use the “dangerous bend” and “double dangerous bend” signs.

The `manfnt` package is no longer enclosed with Refman, it has grown and is now a package of its own.

B.5.3 Seealso

The command `\seealso{n}` prints an arrow and its argument into the left margin. You will find examples for this in the left margin and in chapter 1.

→ Chapter 1

B.5.4 Maxipage environment

The `maxipage` environment is a special kind of `minipage` which extends over the full width of the page. It can be used for long formulas or `tabular` environments. You may use `maxipage` environments inside floats. You cannot use margin notes inside a `maxipage` and no page break will occur while in a `maxipage`. A `maxipage` is always a paragraph of its own with a thick line above and below. You can disable these lines with the `\maxipagerulefalse` command. They are on by default.

The following paragraph is an example for a `maxipage`:

This very long line is an example for a `maxipage`. It extends over the full width of the page, including the left margin.

This is normal text after the `maxipage`.

B.5.5 Fullpage environment

The `fullpage` environment consists of one or more pages where the text extends over the full width of the page. You cannot use margin notes inside a `fullpage` environment. A `fullpage` will always start and end on a page of its own. It may be used for large tables, program listings or anything that does not fit into the normal layout.

→ Page 14

Page 14 is an example for a `fullpage`.

B.5.6 Noparskip

The `\noparskip` removes the vertical space between two paragraphs. It is similar to the `\noindent` command that removes the indent of the first line of a paragraph.

B.5.7 Setleftmarginwidth

The `\setleftmarginwidth` command is no longer supported. You can achieve similar results by using the `\settextfraction` command.

B.5.8 Descriptioncolon

By default a colon is printed after the description label. The command `\descriptioncolonfalse` disables the colon, the `\descriptioncolontrue` re-enables it.

B.5.9 Descriptionleft

The `\descriptionlefttrue` command sets the description label left justified into the margin. The default is right justified which will be achieved with `\descriptionleftfalse`

B.5.10 Maxipagerule

You can disable the rules before and after a `maxipage` with the `\maxipagerulefalse` command and re-enable them with the `\maxipageruletrue` command. The default is on. You should not mix `maxipage`s with and without rules in one document.

B.5.11 Condbreak

The command `\condbreak{2cm}` ensures, that the next 2 cm are either completely on this page or completely on the next. No page break will appear in the next 2 cm.

This is really a hack to achieve what the `\samepage` command often fails to do.

B.5.12 Example

The `example` environment acts like a `verse` environment but uses a `tt` font.

B.5.13 Pageperchapter

The command `\pageperchapter` creates page number that start with 1 for every new chapter. This may be useful for larger manuals. Since it works with chapters it is only available in the `refrep` class.

B.5.14 Smallborder

The normal border around the page is 1 Inch. That is fine for a printed document, but wastes a lot of space when a document is meant for reading on screen. The option `smallborder` reduces the margin to 0.25 Inch.

You can redefine the border with `\setlength\papermarginwidth {0.25in}`. Call `\setpagefraction{0.7}` afterwards to recalculate the page layout.

B.5.15 Dvips

The option `dvips` tells DVIPS about the current page size.

B.5.16 Pdftex

The option `pdftex` tells PDFTEXabout the current page size.

B.5.17 Pagesize

`pagesize` chooses the correct `\special`-command to tell the DVI-driver about the paper size. It works with DVIPS and DVIPDFMX for DVI output and PDFTEX for PDF output.

B.5.18 Ifpdfoutput

You can use `\ifpdfoutput{pdftext}{dvitext}` to write different text depending on the output format. This command was necessary to implement the `pagesize` option and is available for the user as well.

The last four commands have been taken from KOMA-Script, thanks Markus.

Index

attention, 17
attentionsymbol, 17
author, 3

changing the layout, step by step, 7
compositor, 3
condbreak, 19
corporate desing, 4

description environment, 16
descriptioncolon, 18
descriptionleft, 19
design, generic, 5
design, logical, 5
design, visual, 5
designer, 3
dvips, 19

example, 19

footings, 4, 12, 17
footnote, 16
fullpage, 18

headings, 4, 12, 17

ifpdfoutput, 20

layout design, 3
line length, 4
line spacing, 4

manfnt, 17
manual, 4
margin notes, 4, 12
marginlabel, 17
maxipage, 18
maxipagerule, 18, 19
myfootings, 17
myheadings, 17

options, 15

page design, horizontal, 15
page design, vertical, 16
page layout, 4
pageperchapter, 19
pagesize, 19
pagestyle, 17
papermarginwidth, 15
pdftex, 19

refart, invocation, 15
refart.cls, 15
refrep, invocation, 15
refrep.cls, 15
rules, 3
rules of thumb, 4

seealso, 18
setleftmarginwidth, 18
settextfraction, 16
smallborder, 19

telephone directory, 4