For submission to The PracTEX Journal
Draft of July 24, 2008

Page styles on steroids or memoir makes page styling easy

Lars Madsen

Email daleif@imf.au.dk
Website http://home.imf.au.dk/daleif/
Address Department of Mathematical Sciences
Faculty of Science, University of Aarhus
Denmark

Abstract Designing a page style has long been a pain for novice users. Some parts are easy; others need good LATEX knowledge. In this article we will present the *memoir* way of dealing with page styles including new code added to the recent version of *memoir*, that will reduce the pain to a mild annoyance.

We will end the article with a series of common scenarios and how to solve these.

Prerequisites

The prerequisite for this article is access to a recent version of *memoir* (that is *memoir* v1.61803, released in June 2008 or newer; we will refer to v1.61803 as *memoir* 2008).

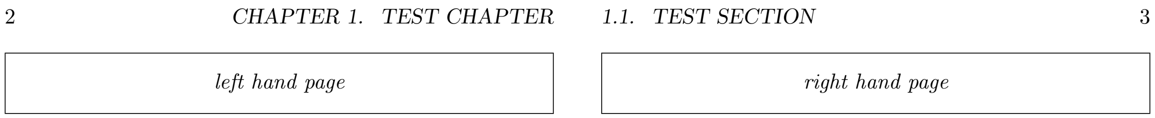
Preliminary information

Examples in this article, that show a graphical result, have been created using the *memoir* class, displaying a two sided setup. The examples will show a left hand (even) page and a right (odd) hand page, with a dummy text block leaving all attention on the headers and the footers. We have also added a dummy chapter, section, subsection and subsubsection, in order to show their effect on the page styles (header information). One can think of the material as page two and three of a chapter that starts out as

```
\chapter{Test chapter}
\section{Test section}
\subsection{Test subsection}
\subsubsection{Test subsubsection}
```

Source code in examples will be shown in a green color; code in the text and showing syntax will be shown in a red color and page styles will be shown in cyan. Here is for instance an example displaying the default page style in *memoir*, known as headings:

```
\pagestyle{headings}
```



In the article we will only handle the *two sided* setup. The *one sided* counterpart comes rather easy; just remember that, whenever the `oneside` option is in effect, the page style design for all pages will be the one corresponding to an odd page.

1 The basics

In this section we will summarise the basics concerning page style handling, plus we will list some of the challenges that we face compared to the standard classes (`book`, `report` and `article`).

1.1 What is a page style?

In LaTeX the *page style* refers to the part of the page building mechanism that attaches the headers and footers to the actual page. Technically, in a two page setup, the headers and footers are inserted via four macros `\@evenhead`, `\@oddhead`, `\@evenfoot` and `\@oddfoot`. Activating a particular page style will redefine these four macros to do whatever the style is designed to.

> Remark. *Please note that the actual vertical placement of the headers and footers are not a matter of the page style. This depends on the dimensioning of the page layout.*

1.2 Activating a page style

A page style can be activated in two ways: globally and locally (here locally means *on this page only*).

```
\pagestyle{⟨style⟩}
\thispagestyle{⟨style⟩}
```

`\pagestyle{`⟨*style*⟩`}` activates ⟨*style*⟩ from the *next page and onwards*, while `\thispagestyle{`⟨*style*⟩`}` enables ⟨*style*⟩ on this page only.

The page style used on a particular page is decided as follows. If any `\thispagestyle` macros have been issued on this page, we use the last one. If no `\thispagestyle` macros have appeared, we use the global page style (i.e. the style recorded to have been set via a `\pagestyle` at the *previous* page break/start).

1.3 Making material available for the headers and footers

In the header and footer we can add static information, i.e. define the four macros to just display some non-changing string. But we also have access to more dynamic information such as the page number (via `\thepage`), and via special macros we can provide access to all sectional titles.

To make titles available we make use of so called *marks*, `\leftmark` and `\rightmark`, which are available on all pages. These can be set, in macros,[1] using

```
\markboth{⟨for left mark⟩}{⟨for right mark⟩}
\markright{⟨for right mark⟩}
```

There is no separate `\markleft`. For the page building mechanism the content of the `\leftmark` and `\rightmark` is determined as follows: `\leftmark` contains the *last* ⟨*for left mark*⟩ issued before the end of the page, while `\rightmark` contains the *first* ⟨*for right mark*⟩ issued on the page. If none have been issued on this page the most recently defined from former pages will be used.[2]

---

[1] The user may also use them manually in the text, though this might annoy copy editors.

[2] Note that in some circumstances some confusion may arise concerning the content of the marks, see [4], section 4.3.4, page 218–219, although the marks work without any problems in most normal chapter based documents.

To ease things, all sectional macros have been equipped with a special macro, named `\⟨sectional macro⟩mark`, e.g. `\sectionmark`, which will be given a title as an argument. These sectional marking macros initially do nothing, but they can be redefined to use `\markboth` or `\markright`.

Remark. *Please recall that the sectional macros in LaTeX supports up to two titles, i.e.*

`\chapter[⟨for head and toc⟩]{⟨for text⟩}`

where ⟨for text⟩ will be used for all if ⟨for head and toc⟩ is missing.

In memoir we extend this even further by supporting the syntax above and supplementing it with

`\chapter[⟨for toc⟩][⟨for head⟩]{⟨for text⟩}`

In addition to the sectional macros, several other macros add material to the marks, including the table of contents (adding `\contentsname`), the index (adding `\indexname`), the bibliography, etc.

1.4 Challenges

As most people know the standard classes (`book`, `article`, `report`), are not particularly flexible. There are basically no configuration interfaces at all; one needs to resort to external packages.

In the area of page styling, the situation in the standard classes is no different. The standard classes have several annoying features regarding the material used in the page styles.

(1) All mark content provided by `\chapter`, `\section`, etc. are hardwired to be in upper case.
(2) Marks set by macros and environments such as `\tableofcontents`, the bibliography and the index are all in upper case. This feature cannot easily be changed without having to copy a large number of code lines and then only changing one or two lines of that code.
(3) In case your setup allows a blank page before starting a new chapter (default in the `book` class), that »blank« page is not really blank; it has the same page style as all other regular pages, which looks odd if there are no other content

on the page. The solution is to manually change `\cleardoublepage` in order to automatically make these pages truly blank.

(4) There is no easy way to redesign the look of the footers and headers.

(5) Creating truly flexible sectional marking macros `\chaptermark`, `\sectionmark`, etc. used internally by `\chapter` etc. demands a lot of LaTeX knowledge from the (novice) user.

The `fancyhdr` package can handle several of these problems but with some costs. As far as I know it does not provide an easy solution to item (5).

2 Page styling in memoir

In *memoir* we deal with two types of page styles, *aliases* and *regular* page styles. An alias page style is a style that just refers to another style, i.e. if the style alias is an alias for the empty style, then `\pagestyle{alias}` is the same as `\pagestyle{empty}`. The advantage of having aliases is that we can, for example, provide the first page of a chapter with its very own page style without the need to create a specific style for it. We just let the chapter style point to some other style, say, plain; and if we later decide that we want a different look for the chapter title page, then we just let chapter point to some other page style.

2.1 How page styles work in memoir

As mentioned earlier the headers and footers are added to the page via the four macros `\@evenhead`, `\@oddhead`, `\@evenfoot` and `\@oddfoot`. In *memoir* a given page style ⟨*style*⟩ consists of a number of specially created macros, all associated with the name ⟨*style*⟩, and activating ⟨*style*⟩ causes LaTeX to rebuild those four macros using the current content of the macros associated with ⟨*style*⟩.

The *memoir* page style interface interacts with the special macros associated to ⟨*style*⟩.

> REMARK. *There is one important thing to remember. Whenever one has changed a page style in memoir (we will cover this shortly), that page style has to be activated in order for the changes to take effect. For a main document style, this means that you have to issue* `\pagestyle{`⟨*style*⟩`}` AFTER *you have made any changes to the style. The reason is that we use the contents of the* ⟨*style*⟩ *macros, not the macros themselves.*

For styles that are just used on local pages, like the chapter style, no reactivation is necessary.

2.2 Creating a new pagestyle

In *memoir* we use one of the following macros to create a new page style:

```
\makepagestyle{⟨name⟩}
\copypagestyle{⟨new name⟩}{⟨old name⟩}
\aliaspagestyle{⟨alias⟩}{⟨original⟩}
```

`\makepagestyle{⟨name⟩}` will create a new page style named ⟨*name*⟩ with all of its corresponding internal macros initialised to doing nothing.

`\copypagestyle{⟨new name⟩}{⟨old name⟩}`, does the same, but initialises the internal macros to use the contents of the internal macros which currently form the style ⟨*old name*⟩. This is very handy, as it allows you to copy a specific style, change the headers in the copy while maintaining the look of the original footers.

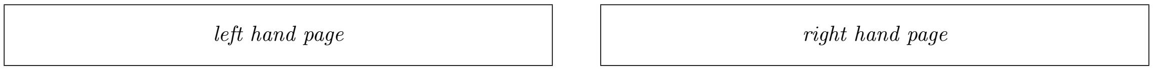
`\aliaspagestyle{⟨alias⟩}{⟨original⟩}` does *not* create internal macros corresponding to its own name. As a consequence, changing an alias page style using the macros we will present in the next section will have unexpected results. The alias uses nothing from ⟨*new name*⟩, so the changes never comes into effect. Basically, do *not* change an alias page style directly; if you need to change an alias style, then first overwrite it (using `\copypagestyle`) with a style you would like to use as a base.

> REMARK. *Note that all three macros will happily* OVERWRITE *an existing page style without warning.*[3]

*   *   *

Here is a list of the regular page styles that are provided by *memoir*:

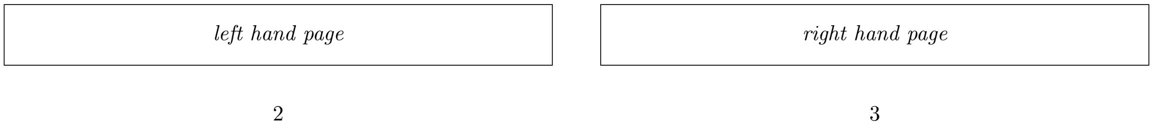
```
\pagestyle{empty}
```



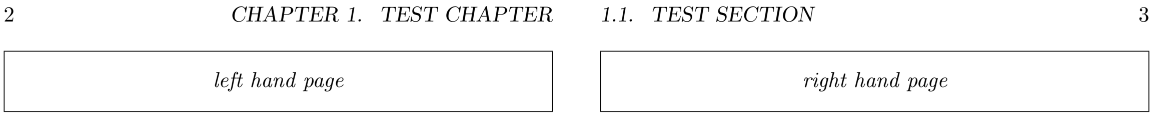
---

[3] A handy feature as we will see later.

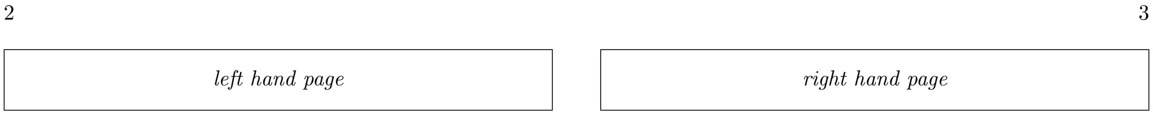
`\pagestyle{plain}`



`\pagestyle{headings}`

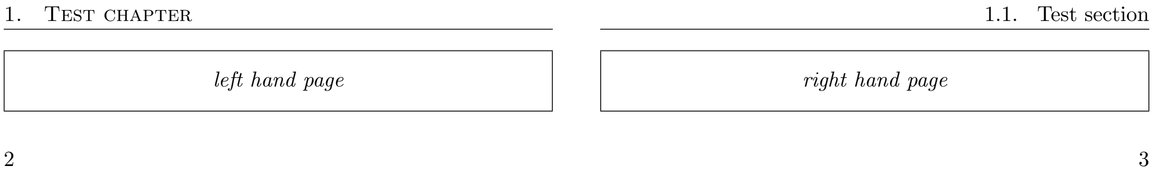


`\pagestyle{myheadings}`

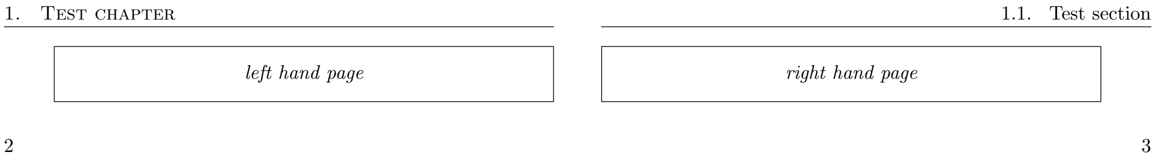


myheadings does not automatically write anything to the headers besides the page number. The user can add the text themselves. This style is handy for writing conference proceedings, where one often sets the headers to show the author names and the title of the contributed article.

`\pagestyle{ruled}`

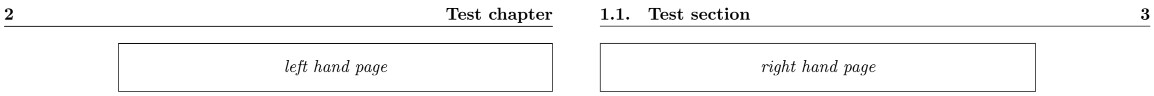


`\pagestyle{Ruled}`



Note how this style protrudes into the outer margin.

```
\setlength{\headwidth}{\textwidth}
\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}
\pagestyle{companion}
```



The companion style mimics the style used in *The LaTeX Companion* by Mittlebach et al, [4]. The design of [4] uses very wide outer margins for notes and such.

* * *

The provided aliases page style in *memoir* are:

chapter	→	plain	title	→	plain
part	→	plain	titlingpage	→	empty
afterpart	→	empty	book	→	empty
cleared	→	empty	afterbook	→	empty
indextitlepagestyle	→	chapter			

The after... styles are the back side of the part and book title pages, see [1]. The indextitlepagestyle has been added in order to make it easy to create very special effects in the index, see for example the index of [3]. title is the page style issued by `\maketitle` whereas titlingpage is the page style for the *first* and *last* page of the `titlingpage` environment.

2.3 Setting up the design of a page style

Designing a page style is a two step process. First there is the visual design, and second there is preparing the macros that provide the information that goes to the headers and footers. In this section we will look at the design part.

The main macros for dealing with the design of the headers and footers are:

```
\makeevenhead{⟨style⟩}{⟨left⟩}{⟨center⟩}{⟨right⟩}
\makeoddhead{⟨style⟩}{⟨left⟩}{⟨center⟩}{⟨right⟩}
\makeevenfoot{⟨style⟩}{⟨left⟩}{⟨center⟩}{⟨right⟩}
\makeoddfoot{⟨style⟩}{⟨left⟩}{⟨center⟩}{⟨right⟩}
```

`\makeheadrule{⟨style⟩}{⟨width⟩}{⟨thickness⟩}`
`\makefootrule{⟨style⟩}{⟨width⟩}{⟨thickness⟩}{⟨skip⟩}`
`\normalrulethickness`

The first four macros are easy to understand. They refer to the headers and footers, where one can place material to the left, in the center or to the right. For example, to create a header which displays titles from the (sub-,subsub-)section in the header for right hand (odd) pages, one could use

```
\makeoddhead{mystyle}{\itshape\rightmark}{}{\thepage}
```

provided that `\⟨(sub-,subsub-)⟩sectionmark` provides information for `\rightmark`.

One simply specifies the style one wants to change, and then specifies the contents for left, center and right for that particular part. Traditionally, for a design that sports chapters, one has the chapter title sent off to the `\leftmark`, whereas all other titles would be sent off to `\rightmark`. In this manner, a widely used setup is:

```
\makeoddhead{mystyle}{\itshape\rightmark}{}{\thepage}
\makeevenhead{mystyle}{\thepage}{}{\itshape\rightmark}
```

The standard headings is like this, it just uses `\slshape` instead of `\itshape`.

To add lines we use either `\makeheadrule` or `\makefootrule`. In most cases the ⟨*width*⟩ will be equal to `\textwidth` and for the ⟨*thickness*⟩ one can use the value set by `\normalrulethickness` (usually 0.4pt) which match most of all other rules set by standard LaTeX. The ⟨*skip*⟩ part will usually be just `\footruleskip`, see [1].

As can be seen with the Ruled and companion page styles, one can also create styles that protrude into the outer margins; these can be set using `\makerunningwidth{⟨style⟩}{⟨width⟩}` and `\makeheadposition{⟨style⟩}{⟨eheadpos⟩}{⟨oheadpos⟩}{⟨efootpos⟩}{⟨ofootpos⟩}` as explained in [1]. Users who would like to use these features are encouraged to have look in [1] and inside the *memoir* source code (`memoir.cls`) to see how these macros work.

2.4 Setting up content providers

As mentioned earlier all sectional macros (`\chapter`, `\section`, etc.) have the ability to make their title available for the page styles via `\chaptermark{`⟨*text*⟩`}`, `\sectionmark{`⟨*text*⟩`}`, etc. which are build into the sectional macros. In addition *memoir* also adds some simple mark macros that are associated with other constructions that may have their own special headers. These include the table of contents (`\tocmark`), list of figures/tables (`\lofmark`), the index (`\indexmark`) and the bibliography (`\bibmark`).[4]

The *memoir* class include an extra function one can use to redefine what the sectional and plain marks should do.

`\makepsmarks{`⟨*style*⟩`}{`⟨*code*⟩`}`

The ⟨*code*⟩ will be executed when ever the ⟨*style*⟩ is activated.

* * *

It has always been hard to create generally applicable sectional marks. One needs to take into account:

1. If sectional numbering for this level is switched off, then no sectional number should be printed in the content provided for `\leftmark` and `\rightmark`.
2. Outside the main matter numbers should never be added.
3. One might want to add some sort of formatting to the number when it is printed, such as adding the word »chapter« in front of the chapter number.

Because of this, a suitable mark for chapters might look like this:[5]

```
\renewcommand\chaptermark[1]{%
  \markboth{\memUChead{%
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \@chapapp\ \thechapter. \ %
      \fi
    \fi
     #1}}{}}%
```

---

[4] Using the `\newlistof` also create a new simple mark.

[5] Slightly altered from the headings definition in *memoir*. `\m@ne` is short for -1 by the way.

How on earth would any novice (or even experienced) user be able to figure this out?

By observing that more or less 90% of all users want dynamic contents within a quite short list of possibilities, we have added some new macros to *memoir* 2008:

```
\createplainmark{⟨type⟩}{⟨marks⟩}{⟨text⟩}
\createmark{⟨sec⟩}{⟨marks⟩}{⟨show⟩}{⟨prefix⟩}{⟨postfix⟩}
\addtopsmarks{⟨pagestyle⟩}{⟨prepend⟩}{⟨append⟩}
\nouppercasemarks \uppercasemarks
```

`\createplainmark` is used to create the plain marks, for example to create a suitable mark for the TOC, one can use

```
\createplainmark{toc}{both}{\contentsname}
```

which would automatically do the same as the more complicated

```
\renewcommand\tocmark{%
  \markboth{\memUChead{\contentsname}}{\memUChead{\contentsname}}}
```

`\createplainmark` is a lot easier to understand. The ⟨*type*⟩ refers to one of `toc`, `lot`,`lof`, `bib`, `index` and `glossary` with obvious meanings.[6]

⟨*marks*⟩ has to be of of »`both`« (meaning ⟨*text*⟩ is added to both the `\leftmark` and the `\rightmark`), »`left`« for just adding to the `\leftmark` (`\markboth{`⟨*text*⟩`}{}`) and »`right`« for just added to `\rightmark` (via `\markright{`⟨*text*⟩`}`). In general you would want to always use `both` for plain marks.[7]

The macro `\memUChead`, used internally, is a *memoir* macro that decides whether to convert the ⟨*text*⟩ in to upper case. The default definition of `\memUChead` is `\MakeUppercase`. One can redefine it as one likes. Though usually one would just issue `\nouppercaceheads` to disable `\memUChead` (i.e. redefine it to doing nothing), alternatively `\uppercaseheads` will make the ⟨*text*⟩ appear in uppercase. To remove the upper case part in a page style it is natural to add `\nouppercaceheads` to the start of ⟨*code*⟩ in `\makepsmarks`.

---

[6] Any use of `\newlistof` will add an extra plain mark.

[7] Otherwise a long bibliography might be missing a header on even pages.

> Remark. *One may wonder why we do not just add the uppercase stuff to the styling for the header? This has something to do with expansion; it has turned out to be better to add it at the marks stage.*

* * *

Setting up the regular plain marks is now as simple as:

```
\createplainmark{toc}     {both}{\contentsname}
\createplainmark{lof}     {both}{\listfigurename}
\createplainmark{lot}     {both}{\listtablename}
\createplainmark{bib}     {both}{\bibname}
\createplainmark{index}   {both}{\indexname}
\createplainmark{glossary}{both}{\glossaryname}
```

The sectional marks are a little bit more complicated. Let us first look at an example. The following code provides the *memoir* standard mark for chapters.[8]

```
\createmark{chapter}{left}{shownumber}{\@chapapp\ }{. \ }
```

Where `\@chapapp` is a *magic* macro defined as

$$\texttt{\textbackslash @chapapp} = \begin{cases} \texttt{\textbackslash appendixname} & \text{if we are within the appendix area} \\ \texttt{\textbackslash chaptername} & \text{else} \end{cases}$$

So by using `\chapapp` we do not have to change the page style when we enter the appendix area.

For `\createmark` the ⟨*sec*⟩ should be one of `chapter`, `section`, `subsection` or `subsubsection`.[9] Whereas ⟨*marks*⟩ is the same as with `\createplainmark`.

⟨*show*⟩ *has* to be either `shownumber` *or* `nonumber`; the effect will be discussed shortly.

The ⟨*prefix*⟩ and ⟨*postfix*⟩ are pre- and postfix texts that goes before and after the sectional number. In the case of chapter marks, ⟨*prefix*⟩ is used to add the `\@chapapp` and some space before the chapter number; and ⟨*postfix*⟩ is used to add a dot and some space after the number.

The content to be provided to the page style by the sectional mark when set up via `\createmark` is determined according to this formula:[10]

---

[8] Though I recommend using `both` instead of `left`.

[9] Similar sectional macros like part, can also be used.

[10] Here *sectional level* corresponds to the `secnumdepth` setting.

```
if ⟨show⟩ equals shownumber
    if current sectional level provides sectional numbers
        if within main matter
            ⟨prefix⟩⟨sectional number⟩⟨postfix⟩
        end if
    end if
end if
⟨the provided title⟩
```

By using these new macros, the style marks for the headings style can now be written

```
\makepsmarks{headings}{%
  \createmark      {chapter} {left} {shownumber}{\@chapapp\ }{. \ }
  \createmark      {section} {right}{shownumber}{}           {. \ }
  \createplainmark {toc}     {both} {\contentsname}
  \createplainmark {lof}     {both} {\listfigurename}
  \createplainmark {lot}     {both} {\listtablename}
  \createplainmark {bib}     {both} {\bibname}
  \createplainmark {index}   {both} {\indexname}
  \createplainmark {glossary}{both} {\glossaryname}
}
```

Wasn't that simple? It is easy to write and easy to understand—a clear improvement over earlier methods.

> REMARK. *What is up with the ⟨show⟩ option? What is this good for? It has been added to enable users who have, say, subsection numbers activated, but who do not want the subsection number to appear in the header. They should just use* `nonumber`*.*

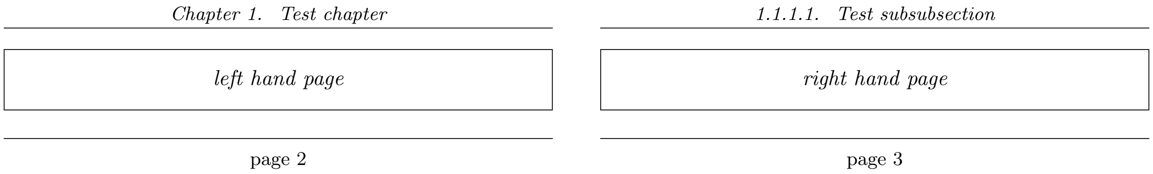
The last new macro, `\addtopsmarks{⟨pagestyle⟩}{⟨prepend⟩}{⟨append⟩}` can be used to extend a given `\makepsmarks{⟨style⟩}`. The prime example is to extend, say, headings to also include marks for sub- and subsubsection.

```
\addtopsmarks{headings}{}{
  \createmark{subsection}   {right}{shownumber}{}{. \ }
  \createmark{subsubsection}{right}{shownumber}{}{. \ }
}
```

2.5 An example

As s final example let us create a style like the ruled style from scratch. We would like to have the text in the center both in the header and the footer.

```
\makepagestyle{myruled}
\makeheadrule {myruled}{\textwidth}{\normalrulethickness}
\makefootrule {myruled}{\textwidth}{\normalrulethickness}{\footruleskip}
\makeevenhead {myruled}{}{\small\itshape\leftmark} {}
\makeoddhead  {myruled}{}{\small\itshape\rightmark}{}
\makeevenfoot {myruled}{}{\small page \thepage}    {}
\makeoddfoot  {myruled}{}{\small page \thepage}    {}
\makeatletter % because of \@chapapp
\makepsmarks  {myruled}{
  \nouppercaseheads
  \createmark      {chapter}      {both} {shownumber}{\@chapapp\ }{. \ }
  \createmark      {section}      {right}{shownumber}{}           {. \ }
  \createmark      {subsection}   {right}{shownumber}{}           {. \ }
  \createmark      {subsubsection}{right}{shownumber}{}           {. \ }
  \createplainmark {toc}          {both} {\contentsname}
  \createplainmark {lof}          {both} {\listfigurename}
  \createplainmark {lot}          {both} {\listtablename}
  \createplainmark {bib}          {both} {\bibname}
  \createplainmark {index}        {both} {\indexname}
  \createplainmark {glossary}     {both} {\glossaryname}
}
\makeatother
\setsecnumdepth{subsubsection}
\pagestyle{myruled}
```



2.6 Conclusions

In section 1.4 on page 4 we listed some challenges; here are our conclusions to those challenges:

ad (1): The automatic upper casing feature has been factored out. Via `\createplainmark` and `\createmark`, the sectional marks, e.g. `\chaptermark`, will internally use `\memUChead` as a wrapper macro around the text to be handed to the headers. `\memUChead` can be defined to use `\MakeUppercase` (the default) using `\uppercaseheads` or redefined to doing nothing via `\nouppercaseheads`. You could even redefine `\memUChead` to something completely different.

ad (2): In *memoir*, these settings have been separated out; and they will be set using plain mark macros such as `\tocmark`, `\bibmark` and `\indexmark`, all of which can now easily be set using `\createplainmark`.

ad (3): In *memoir*, `\cleardoublepage` (and similar macros) have been designed to issue the cleared page style on *blank* pages. This style *alias* initially points to empty but can be redefined to whatever the user wants it to be.

ad (4): The look of the headers and footers can easily be redefined using the *memoir* interface presented in section 2.3 on page 8.

ad (5): For the vast majority of all users the `\createmark` macro will provide all the flexibility users will ever need.

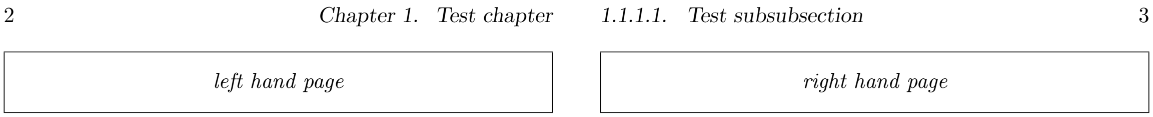
3 Scenarios

In this section we will list some (possible) real life scenarios and their solutions.

> Scenario 1. *The headings style is* ok*, but we also want sub- and subsubsection titles in the headers, and please drop the all upper case thing.*

No problem

```
\addtopsmarks{headings}{%
  \nouppercaseheads % added at the beginning
}{%
  \createmark{subsection}   {right}{shownumber}{}{. \space}
  \createmark{subsubsection}{right}{shownumber}{}{. \space}
}
% use the new settings
\pagestyle{headings}
\setsecnumdepth{subsubsection} % activating subsubsec numbering in doc
```



Scenario 2. *Requirements states that possible 'blank' page before at new chapter, it should be in the normal page style.*

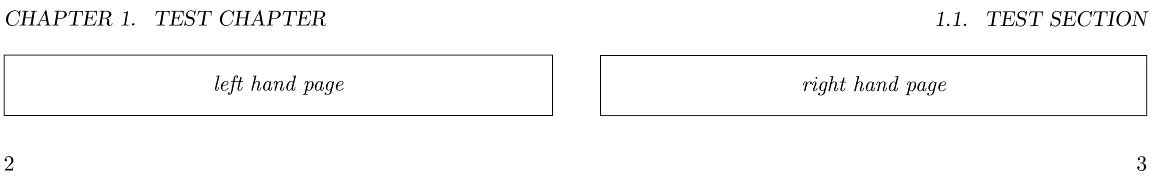
Assuming that headings is the style used

```
\aliaspagestyle{cleared}{headings}
```

Scenario 3. *Create a style where the header is flushed towards the outer edge of the text block, containing the header information. Do the same for the footer which should contain the page number. Additionally all other used pages styles should have their footer set in the same manner.*

The page style itself is straight forward; the 'hard' part is the requirement on the other used styles. But wait; in the standard setup all auxillary page styles that use page numbers are all aliases to plain or are the plain style itself, so it suffices to alter plain. To save a bit of space in this article we will base the new page style on headings and just use the marks that come with headings; insert your own replacements.

```
\copypagestyle{outer}{headings}
\makeoddhead{outer}{}{}{\slshape\rightmark}
\makeevenhead{outer}{\slshape\leftmark}{}{}
\makeoddfoot{outer}{}{}{\thepage}
\makeevenfoot{outer}{\thepage}{}{}
% fix plain
\copypagestyle{plain}{outer} % overwrite plain with outer
\makeoddhead{plain}{}{}{}    % remove right header
\makeevenhead{plain}{}{}{}   % remove left header
\pagestyle{outer}
```



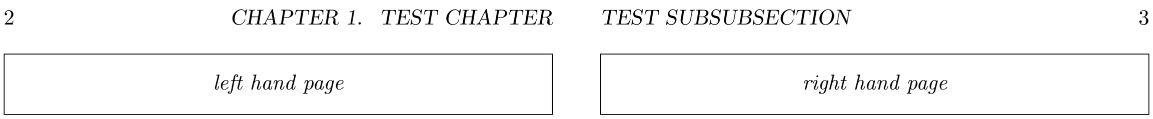
Scenario 4. *On pages with only floats we would like to have no headers or footers.*

Here *memoir* has a trick up its sleeve: `\mergepagefloatstyle`; see [1], page 174, again assuming headings as the regular style used.

```
\mergepagefloatstyle{mergedstyle}{headings}{empty}
\pagestyle{mergedstyle}
```

Scenario 5. *Look of headings is* OK, *but please remove the section numbers from the header.*

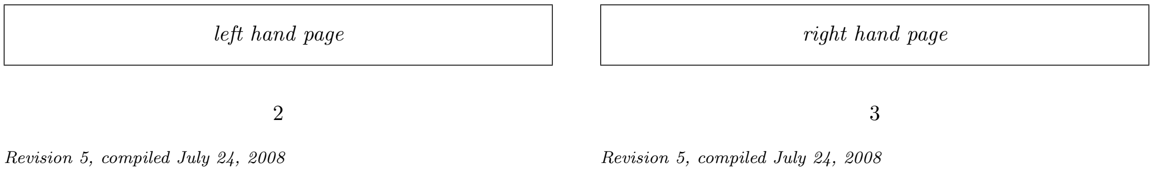
```
\addtopsmarks{headings}{}{
  \createmark{section}      {right}{nonumber}{}{}
  \createmark{subsection}   {right}{nonumber}{}{}
  \createmark{subsubsection}{right}{nonumber}{}{}
}
\pagestyle{headings}
```



Scenario 6. *Add a the revision number to all pages.*

There are several ways to do this; the easiest way is to use the `eso-pic` package and use it to place the revision information on to the background. But we can also solve it using page styles. This requires creating a new page style (or modifying an existing one) and a rebuild of plain. The construction can easily be combined with other constructions to extract Subversion headers; see [2]. We will place the revision information below the footer to the left. We will modify headings but will show plain pages, so as to show the placement relative to the footer.

```
\newcommand\AddRevision{%
   \setlength\unitlength{1mm}
   \begin{picture}(0,0)
     \put(0,-7){\footnotesize\itshape Revision 5, compiled \today}
   \end{picture}}
\makeoddfoot{headings} {\AddRevision}{}{}
\makeevenfoot{headings}{\AddRevision}{}{}
\makeoddfoot{plain}    {\AddRevision}{\thepage}{}
\makeevenfoot{plain}   {\AddRevision}{\thepage}{}
%\pagestyle{headings}
\pagestyle{plain}
```

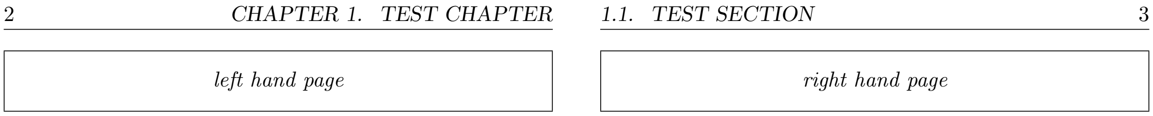

left hand page
right hand page
2
3
Revision 5, compiled July 24, 2008
Revision 5, compiled July 24, 2008

This could be extended in such a way that if, say, all chapters are in files of their own, then revision data for each separate chapter can be written at the bottom of the first page of each chapter. This is simply done by altering the chapter style, as above.

A colleague used the trick of using page styles to add other contents to the document, to implement a LaTeX version of our university letter style. It adds a colophon on the first page of a letter.

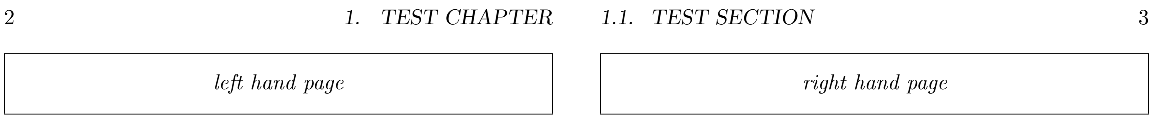
Scenario 7. *Add a line under the header.*

```
\makeheadrule{headings}{\textwidth}{\normalrulethickness}
\pagestyle{headings}
```



Scenario 8. *Remove the »Chapter« from the headings style.*

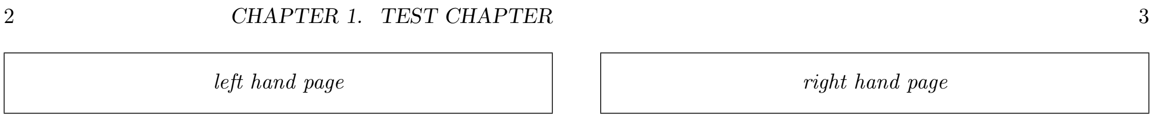
```
\addtopsmarks{headings}{}{
  \createmark{chapter}{both}{shownumber}{}{. \space}
}
\pagestyle{headings}
```



Scenario 9. *Remove the marks set by* `\section`.

This is a quite unusual request, so there is no nice interface for it. The easiest solution is

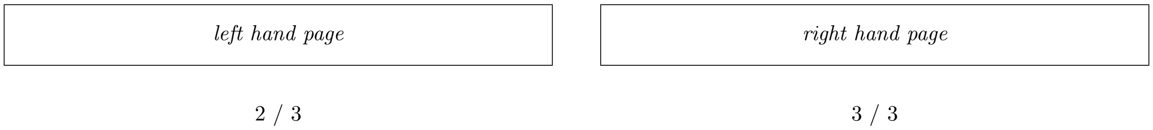
```
\addtopsmarks{headings}{}{%
  \renewcommand\sectionmark[1]{}
}
\pagestyle{headings}
```



Scenario 10. *I'd like to have a »page / total pages« footer. How do I implement that?*

In *memoir* we record two *total* numbers: the total number of sheets in the document, and the page number of the very last page. These are available via `\thelastsheet` and `\thelastpage`. Note that they are generally not the same, as the page number is usually reset to 1 at the beginning of the main matter. Let's change plain to solve the scenario:

```
\makeevenfoot{plain}{}{\thepage\ / \thelastpage}{}
\makeoddfoot {plain}{}{\thepage\ / \thelastpage}{}
\pagestyle{plain}
```



References

[1] Peter Wilson, *The Memoir Class for Configurable Typesetting – User Guide*, seventh edition, The Herries Press, 2008.

[2] PracTeX Journal, issue 2007-3. Available at `http://www.tug.org/pracjourn/2007-3/index.html`

[3] Lars Madsen, *Introduktion til LaTeX*, preliminary version of the third edition, 2008. Available via `http://www.imf.au.dk/system/latex/bog/`.

[4] Frank Mittelbach and Michel Goossens, *The LaTeX Companion*, Addison-Wesley, 2nd edition, 2004. ISBN 0-201-36299-6. With Johannes Braams, David Carlisle and Chris Rowley and contributions by Christine Detig og Joachim Schrod.