Headers and footers with titleps*

Javier Bezos†

2019/10/16

Contents

1. Introduction

The titleps package provides tools for one-stage setting of page styles (headlines and footlines). A higher-level interface is used, where the mark mechanism is hidden and there is no need to deal with \leftmarks and \rightmarks – just use a command or variable registered as a “mark” as the expected value will be returned, i.e., those when the mark was emitted, either by a sectioning command or explicitly with \chaptermark, \sectionmark, etc. A simple example, whose meaning should be obvious, is:

\newpagestyle{main}{
    \sethead[\thepage]\[\chaptertitle]\[(\thesection\) % even
    \{\thesection}\]\[\sectiontitle]\[\thepage\]} % odd
\pagestyle{main}

Other features are:

• Working top marks, compatible with floats (unlike the standard \topmark, which does not work correctly in \LaTeX).

• Access to top, first and bottom marks in a single headline/footline (e.g., the first and last section numbers).

• Marks for more than 2 sectioning levels.

• Auto-defining of suitable \sectionmark and the like.

• Simple (and not so simple) headrules and footrules.

• Headlines and footlines for pages with floats.

• Headlines and footlines for specific floats (a sort of \thimestep for floats).

• Multiple sets of marks (named here marksets and extra marks).

It can be used without titlesec, but you will get most of it when used together. To load it as a separate package, use the customary \usepackage{titleps}, but with titlesec you have to load it with:

\usepackage[pagestyles]{titlesec}

*The titleps package is currently at version 2.13. © 2016 Javier Bezos. All Rights Reserved.
†For bug reports, comments and suggestions go to [http://www.texnia.com/contact.html](http://www.texnia.com/contact.html) English is not my strong point, so contact me when you find mistakes in the manual. Other packages by the same author: gloss (with José Luis Díaz), unnatural, accents, tensind, esindex, dotlessi.
Specific tools for sectioning command marks are provided, which are considered the main markset (unnamed) and carry out some internal task. However, titleps can be used as well for headers with the first an last entry of a page in a dictionary or index, for example, thank to the extra marks described below.

It works with standard classes and many others; however, additional \LaTeX layout adjustments could be necessary in some of them (for instance the AMS classes) and several packages emitting marks directly with \markboth and \markleft.

Once loaded the package, you can define your own page styles and then activate them, but note page styles so defined cannot be used with \thispagestyle except if the surrounding page style has been defined with titleps, too.

## 2. Defining Page Styles

\begin{itemize}
  \item \texttt{\newpagestyle{\langle name\rangle}\{\langle global-style\rangle\}\{\langle commands\rangle\}} \texttt{\renewpagestyle{\langle name\rangle}\{\langle global-style\rangle\}\{\langle commands\rangle\}}

  Defines a new style or redefines an existing one, named \langle name\rangle. For convenience, spaces and end of lines are ignored, and you need not “hide” them with %.

  \langle global-style\rangle is any command to be applied to both footlines and headlines. Fragile commands are allowed.

  In \langle commands\rangle you can use:

  \begin{itemize}
    \item \texttt{\sethead\{\langle even-left\rangle\}\{\langle even-center\rangle\}\{\langle even-right\rangle\}\{\langle odd-left\rangle\}\{\langle odd-center\rangle\}\{\langle odd-right\rangle\}}
    \item \texttt{\setfoot\{\langle even-left\rangle\}\{\langle even-center\rangle\}\{\langle even-right\rangle\}\{\langle odd-left\rangle\}\{\langle odd-center\rangle\}\{\langle odd-right\rangle\}}
  \end{itemize}

  Set the corresponding part in headlines and footlines (note the “visual” layout of parameters). The optional arguments are the three or none. If no optional arguments are given the odd-page settings applies to both even and odd pages. A couple of starred variant (\setfoot*, \sethead*) reverses the order of the settings for even pages (obviously, with them the optional arguments are not allowed). If \sethead is not given, then the headline it is left empty, and same for \setfoot and the footline.

  In the arguments of \sethead/\setfoot and because of its one-stage mechanism, we must distinguish between two groups of commands:

  \begin{itemize}
    \item The first one includes those saved when the marks are emitted, with information related to sections, and includes:
      \begin{itemize}
        \item \texttt{\thechapter, \thesection, \thesubsection...}
        \item \texttt{\chaptertitle, \sectiontitle, \subsectiontitle...} which print the chapter, section... title.
        \item (Only titlesec. Not in titleps.) \texttt{\ifthechapter{\langle \langle true\rangle\{\langle false\rangle, \langle \langle true\rangle\{\langle false\rangle, \langle \langle true\rangle\{\langle false\rangle...} which expands to \langle true\rangle except if the corresponding title lacks of label or there is no title yet after the superior level (for example, between \chapter and a subsequent \section).
        \item Any other command or value “registered” as “mark.” (See below.)
      \end{itemize}

      They are ordinary commands, with no internal marks at all, and you can test if, for example, they are empty.
    \item The second group are those expanded on every page, and includes:
      \begin{itemize}
        \item \texttt{\thepage.}
        \item Any other command not included in the previous items.
      \end{itemize}
  \end{itemize}
\end{itemize}
Sets which \ldots title commands are to be defined, and when the marks are emitted; any number of levels is allowed (1, 2, 3...). For example \settitlemarks{chapter,section} means that:

- \chaprttitle and \secttitle are the allowed titles in headlines,
- \secttitle is reset in \chapter,
- (Only titlesec. Not in titlesps.) if\thechapter and if\thesection are valid tests.
- Marks are updated in \chapter and \section.

Default is \settitlemarks{chapter,section}, but \settitlemarks{section,subsection} in article class. \settitlemarks may be used outside \renewpagestyle to set the default to be used in newly defined and redefined page styles – it is used in \pagestyle.

The starred version \settitlemarks* creates parallel extra marks as described below.

If you want a rule below the headline and above the footline. You may also set their width directly with the commands \setheadrule and \setfootrule. (For instance, \setheadrule{.4pt} which is, by the way, the default value.)

These commands are local to a page style, i.e., defining them directly in the preamble will not work because the page style settings will override them.

These commands are used by titlesps to typeset the rules. If there is no rule, both commands are empty (this value is the default). \setheadrule(dim) just stands for

\renewcommand{\makeheadrule}{% \makebox[0pt][l]{\rule[-.3\baselineskip]{\linewidth}{\linewidth}{(dim)}}%

unless if dim is 0 pt, which empties \makeheadrule (and similarly \setfootrule).

You can access the total header/footer width with \linewidth but the box enclosing the rule is in fact dimensionless. Hence, you must be concerned with the rule placement only. Its baseline is the same as that of the header/footer. That means the material to be used as rule should be raised or lowered with \raisebox, the lifting argument in \rule or by using sensible coordinates in picture, for instance.

That makes easy to place material above and below. For instance, the following code creates a headline with a black bold line above and a red rule below (the color package is required):

\renewcommand{\makeheadrule}{{% \makebox[0pt][l]{\rule[-.3\baselineskip]{\linewidth}{0.8pt}}}% \color[named]{Red} \rule[-.3\baselineskip]{\linewidth}{0.4pt}}%

Of course, the material to be used as rule is not restricted to actual rules; pictures and leaders, for instance, are allowed, too.

3. On \markboth and \markleft

As The B\TeX\ Companion explains “Leslie Lamport had a specific marking scheme in mind when he designed those commands”. They are clearly intended for the heading and myheading pagestyles, with a main-mark on the left page and a sub-mark on the right one.

In order to provide the functionality offered by titlesps, a more general, less specific mechanism is devised, which means \markboth here does not make sense, because it doesn’t work like in standard B\TeX, and in fact, can lead to unpredictable results.

\setmarks is deprecated. Now \setmarks{<primary>}{<secondary>} is \settitlemarks{<primary>, <secondary>}
Therefore, using directly the marking commands is discouraged (see titlesec.pdf), but if you need them, you may write:
\chapter*{My Chapter}
\chaptermark{My Chapter}

But some issues remain, because there are a number of packages assuming the pagestyle is heading or myheading. This is, of course, a reasonable guess, but not always correct (even if titleps is not used at all). Most of them do the right thing -- \markboth is encapsulated into a macro easily redefinable so that the user can modify it if necessary. This is true, for example, for biblatex, but unfortunately not for endnotes. When this happens, you can resort to the following macro.

\setmarkboth{(code-to-use)}

From this point on, \markboth behaves as defined. It doesn't redefine directly the command, but rather provides a replacement code (you can use \#1 and \#2 for the two arguments). With \resetmarkboth (which can be used inside \setmarkboth) the original definition will be again in force. For example, to cancel the next \markboth:
\resetmarkboth\markboth{}

4. **Headline/footline width**

    \widenhead[⟨even-left⟩][⟨even-right⟩][⟨odd-left⟩][⟨odd-right⟩]
    \widenhead*[⟨even-right/odd-left⟩][⟨even-left/odd-right⟩]

Makes the headlines/footlines lines wider. The extra width is asymmetrically added and hence the two [four] arguments. Like \sethead, the starred version reverses the setting for even pages—for example, \widenhead*[0pt]{6pc} is the same as \widenhead{6pc}{0pt}{6pc}{0pt}.

To be used outside \(\texttt{re}newpagestyle\).

5. **Marks**

Marks can be retrieved in 4 ways:

**top marks** If a page begins with text, they are the marks belonging to the sectioning unit of that text. If a page begin with a sectioning title, they are the marks of that title. In other words, top marks refer to the section immediately adjacent to the headline.

**first marks** The marks of the very first sectioning title.

**bottom marks** The marks of the very last (bottom) sectioning title (also known as bot marks).

**next top marks** A bottom mark with the values of the top mark of the next page (only \texttt{straight} class or equivalent); by picking some value from it and comparing it with the corresponding value from the bottom marks you may know if a section continues in the next page. Of course, it will not be used very often.
Standard \LaTeX{} provides only first and bottom marks. As you see, with \texttt{titleps} you have two additional marks, namely, top and next top marks.

It should be stressed top marks are not the same as the primitive \texttt{\topmark} (which, in fact, is not used internally at all). They are true top marks, returning the expected result when the title is at the top of the page. Further, they are compatible with floats, which is not true with \texttt{\topmark}s. However, they fail with explicit page breaks, but you can use \texttt{\pretitlemark} as described below.

\begin{verbatim}
outertitlemarks innermarks topmarks botmarks (package options)
\end{verbatim}

innermarks is the default in \LaTeX{}, with bottom marks in even pages and first marks in odd pages. More convenient is outermarks, with top marks in even pages and bottom marks in odd pages; it is the more usual system in classic scientific literature and it is described in \emph{The \TeX{}book}, p. 259. Both of them are intended for two side printing; topmarks/botmarks are intended for one side printing with top/bot marks in every page (they can be used in two side printing as well).

\begin{verbatim}
\bottitlemarks \toptitlemarks \firsttitlemarks \nexttoptitlemarks
\outertitlemarks \innertitlemarks
\end{verbatim}

These commands set which mark the values of the subsequent \texttt{\thesection}, \texttt{\sectiontitle}, etc., will be taken from. These macros are switches, and therefore subsequent commands and variables are taken from the selected mark. The following is valid:

\begin{verbatim}
\toptitlemarks\textbf{\thesection.} \sectiontitle --
\bottitlemarks\textbf{\thesection.} \sectiontitle
\end{verbatim}

You can use them freely in your headers: just for fun, the following header shows the section label from three of these marks:

\begin{verbatim}
\newpagestyle{funny}{
 \sethead{\toptitlemarks Top is \thesection}{\firsttitlemarks First is \thesection}{\bottitlemarks Bot is \thesection}
}
\end{verbatim}

In multilingual documents, the language in force is that corresponding to the package options — command versions does not switch the language, but you can do it easily with the help a new title mark and then switching the language, as explained below.

\begin{verbatim}
\newtitlemark{(command-name)}
\newtitlemark*[\texttt{\variable-name}]
\end{verbatim}

Add a command or a variable to the list of “marks” to be emitted at a sectioning command. The macro must be parameterless, and the variable is either a length, if it has the form \texttt{\langle name\rangle}, or a counter, if it has the form \texttt{(name)} (i. e., a string). For example, if you write (with babel):

\begin{verbatim}
\newtitlemark{\languagename}
\end{verbatim}

then a \texttt{\languagename} in the head/foot will contain the language when the corresponding mark was emitted; unfortunately, babel does not allow \texttt{\languagename} as the first argument of \texttt{\foreignlanguage} (it enters in a infinite loop), but there is a simple trick – just use an intermediate command:

\begin{verbatim}
\newcommand{\titlelanguage}{\languagename}
\newtitlemark{\titlelanguage}
\end{verbatim}

\footnote{\texttt{outertmarks} may not work well in conjunction with explicit \texttt{\section} mark commands \texttt{\section} mark commands
\footnote{Not exactly: top marks must not be used in the page where a chapter begins, except by overriding the default definition and making it of \texttt{top} class.}
(What is saved is the expansion of \titlelanguage, which is turn is the expansion of \languagename. Alternatively, you can create the intermediate command with \let in the header itself.)

Marks to be emitted at other places (not sectioning commands) are handled with extra marks and marksets (see below).

Commands must be robust, but not protected, or the value returned will not be the right one. Remember there are two kinds of robust commands – those containing in turn robust commands and those protected with, say, \DeclareRobustCommand. An example of how to use protected commands is a layout used sometimes if sections are long as well as their titles, where the title is splitted across the left and the right pages. Here is how this can be carried out:

\DeclareRobustCommand\splitheader[2]{#1 #2}
\def\contsplitheader{}
\def\dosplitheader{\gdef\contsplitheader{}\DeclareRobustCommand\splitheader[2]{##1\gdef\contsplitheader{##2}}}
\renewpagestyle{headings}{\sethead[\thepage]\[
\dosplitheader\sectiontitle\]
{\contsplitheader}
{\thepage}\}
\pagestyle{headings}

(Note the header also uses \DeclareRobustCommand.)

\pretitlename\section\text{Performing Brahms}
\newpage
\section{Performing Brahms}

Note the text in the premark is the title of the next section. The counter is advanced temporarily, so that it has the right value, but if for some reason you do not want that, just use \pretitlename.*

\ifsamemark\topmarks\thesection\true\false

One of the disavantaged of the switching mechanism is that comparing a command from, say, the top marks and the bottom marks is not trivial. This macro just makes it simpler. If you want to compare the current \thesecount with that in the top mark, write:

\ifsamemark\topmarks\thesecount\true\false

\footnote{Unfortunately, \DeclareRobustCommand has a misleading name, because a command created with \newcommand can be robust, too. A better name would have been \DeclareProtectedCommand.}
6. Running heads with floats

\texttt{psfloats} (package option)

This package option activates the commands described in this section.\footnote{It redefines some internal \LaTeX{} commands and there could be incompatibilities with other packages modifying the same commands. That’s the very reason these commands are loaded only optionally. As of version 2.6 the syntax was extended with a new argument to allow extra stuff in running heads/foots. This change is backward compatible—just preserve the old name floatps, which is deprecated, if you want the pre-2.6 behaviour.}

\input{code/6_1_running_heads_with_floats}

The arguments with a dot are similar to those of \texttt{\sethead}. The final argument says the head is used when there is a float of the specified types (default is \texttt{tp} in headers, and \texttt{bp} in footers). For example:

\begin{verbatim}
\newpagestyle{main}{\sethead ... your definition \setfoot ... your definition \headrule \footrule \setfloathead{}{}{}{\setheadrule{0pt}}[p] \setfloatfoot{}{}{}{\setfootrule{0pt}}[p]}
\end{verbatim}

removes the header/footer on float pages, including the rules.

\begin{verbatim}
\nextfloathead*{}{}{}{}[t]
\begin{figure}
...\end{figure}
\end{verbatim}

You may force a header (respectively footer) when a certain float is the first top one (respectively last bottom) with those commands just before the corresponding float, where the arguments with a dot are similar to those of \texttt{\sethead}. Those commands are used by themselves, without putting them inside \texttt{\renewpagestyle}. So, to empty the headline when a certain figure is placed at the top (because it is a bleed, for example):

\begin{verbatim}
\nextfloathead{}{}{}{}[t]
\begin{figure}
...\end{figure}
\end{verbatim}

7. Extra marks

Both \texttt{titlesec} and \texttt{titleps} provide tools for handling extra marks, if you are using an \TeX{} based engine (currently most of them) and load the \texttt{etex} package or equivalent. The way they work are conceptually similar to the title marks described above – you set which commands or variables are to get synchronized in heads and foots. To use extra marks, you have to load them with the following package option:

\texttt{extramarks} (package option)

Extra marks are subsidiary. If page styles are modified internally by another package and the document does not make use of sectioning commands, you might still need an empty title mark (like \texttt{\sectionmark{}}), as well as to synchronize the language by hand.

In many cases, the following “short” marks mechanism will be enough. With it, you simply say with \texttt{\newshormark} the name of a command to be used in heads or foots, and then emit the mark at appropriated places (short marks for variables are not available). An example illustrates how this is carried out:
\newcommand\lemmatitle{}
\newcommand\lemma[1][]{%  \renewcommand\lemmatitle{#1}%  \preshortmark\lemmatitle \iff you want top marks  \pagebreak[2]%  \vspace{lem}%  \textbf{#1}\enspace  \shortmark\lemmatitle}  
\newpagestyle{main}{%  \sethead{}{\topshortmark\lemmatitle  \--\firstshortmark\lemmatitle  \--\botshortmark\lemmatitle}{}%  }%  \pagestyle{main}  

Top marks require emitting the marks before the page break – just use the command \preshortmark as shown above. Otherwise, you may omit it. Unlike title marks and extra marks, the selector of short marks (\topshortmark, \firstshortmark, \botshortmark, \nexttopshortmark), is not a switch and must be always immediately followed by the command to be retrieved.

For a more general approach use the following tools.

\newmarkset{⟨markset⟩}

A markset contains a set of related commands and variables intended to be emitted simultaneously as marks, so that their values can be retrieved in heads/foots. A typical markset could be a title and its counter (and perhaps, in a multilingual document, the active language, i.e., \languagename in babel, or an intermediate command). There is always a special unnamed markset for sectioning marks, which is the main one and handled with the tools described above (with title in their names).

\newextramark{⟨markset⟩}{⟨macro-name⟩}
\newextramark*[⟨markset⟩]{⟨variable-name⟩}

Add to a markset a command or a variable to be synchronized. They are identical to \newtitlemark and \newtitlemark*, except for the additional parameter with the markset name.

\botextramarks{⟨markset⟩} \topextramarks{⟨markset⟩} \firstextramarks{⟨markset⟩} \nexttopextramarks{⟨markset⟩} \outerextramarks{⟨markset⟩} \innerextramarks{⟨markset⟩}

The extra counterparts of the title commands described above, with the same behaviour. For example:

\newcommand\lemmatitle{} \newcounter{lemma}
\newmarkset{lemma}
\newextramark{lemma}{\lemmatitle}
\newextramark*[lemma]{lemma} \iff ie, the counter 'lemma'
\newcommand\lemma[1][]{%  \renewcommand\lemmatitle{#1}%  \stepcounter{lemma}%  \preeextramark\lemmatitle \iff \pagebreak[2]%
By default, \LaTeX{} and \texttt{titleps} use a single \textit{markset} for all sectioning levels. This is problematic, for example, in the \texttt{article} class, because sections does not start a new page. Consider the following piece of code:

\section{A section}
\subsection{A subsection}

On odd pages with first marks and subsection titles, the head has no title, because \texttt{\subsection} is the second mark, not the first one.

To overcome this limitation, you can use the starred version of \texttt{\settitlemarks}, which creates parallel \textit{marksets} (one per level) synchronized with the main \textit{markset}. They are retrieved like other extra marks, but they are special in that you cannot emit them by yourself – they are always emitted by the corresponding sectioning mark. So, with

\settitlemarks*{(\texttt{section}),\texttt{subsection})}

and a page style containing

\firstextramarks{subsection}\texttt{\subsectiontitle}

you retrieve the mark emitted by the first \texttt{\subsection}, even if preceded by a \texttt{\section}. (Top marks are best retrieved from the main \textit{markset}.)

They are extra marks, and therefore language synchronization has to be done by hand – add \texttt{\languagename} or an intermediate command to the main \textit{markset} with \texttt{\newtitlemark} and, if you want, define a command. For example:

\begin{verbatim}
% use an intermediate command because we'll need \foreignlanguage\newcommand\titlelanguage{\languagename}\newtitlemark{\titlelanguage}

% define a command to ease defining page styles\newcommand{\toplang}[1]{% \topextramarks{section}{#1}}\foreignlanguage{\titlelanguage}{#1}}\end{verbatim}

\section{A couple of examples}

Because of the way marks are handled, special constructs like this are allowed:

\begin{verbatim}
\newpagestyle{main}\lsmall\sffamily{\sethead{\textbf{\thepage}}{[[\texttt{\chaptertitle}]\texttt{[\texttt{\sectiontitle}]}}{[[\texttt{\sectiontitle}--\texttt{\bottitlemarks}\texttt{\thesection}]\texttt{[\texttt{\sectiontitle}--\texttt{\bottitlemarks}\texttt{\thesection}]}}}}\textbf{\thepage}))))
\end{verbatim}
As you can see, the range of sections in a page with \thesection is printed in both left and right heads. Of course, this example should be fine tuned to collapse the ranges if there is a single section in a page (with \ifsamemark), but it gives a hint of how to get stunning headers.

In this document, the following styles are used:

\widenhead{2.1pc}{0pc}
\renewpagestyle{plain}{\small\sffamily\slshape}{\footrule}{\setfoot{}{\thepage}{}}
\newpagestyle{myps}{\small\sffamily\slshape}{\headrule}{\sethead{titlesec -- titleps}{\sectiontitle}{\thepage}}

The following definitions provide page styles similar to those in Lamport’s *The \LaTeX book* (with the calc package):

\widenhead*{0pt}{\marginparsep + \marginparwidth} % symmetrically
\renewpagestyle{plain}{}
\newpagestyle{latex}{\bfseries}{\headrule}{\sethead{\thepage}{\chaptertitle}{\thesection \ \sectiontitle}{\thepage}}
\pagestyle{latex}