The **titling** package[*]

Peter Wilson
Herries Press
`pandgwilson at earthlink dot net`

2004/09/26

Abstract

The titling package provides control over the typesetting of the `\maketitle` and `\thanks` commands. The values of `\title`, `\author` and `\date` are also retained, and there may be mutiple titles in a document. New titling elements may be defined for printing by `\maketitle`.

Contents

1 Introduction

The format for typesetting the `\maketitle` command is fixed in the standard LaTeX classes. The titling package provides some controls for modifying the appearance of the titling information presented via `\maketitle`.

This manual is typeset according to the conventions of the LaTeX DOC-STRIP utility which enables the automatic extraction of the LaTeX macro source files [GMS94].

Section 2 describes the usage of the titling package and commented source code is in Section 3.

---

[*]This file (`titling.dtx`) has version number v2.1c, last revised 2004/09/26.

2   The **titling** package

The typeset format of the \maketitle command is virtually fixed within the LATEX standard classes. The titling package provides a set of formatting commands that can be used to modify the appearance of the titling information; that is, the contents of the \title, \author and date commands. It also keeps the values of these commands so that they may be printed again later in the document.

The package inhibits the normal automatic cancellation of titling commands after \maketitle. This means that you can have multiple instances of the same, or perhaps different, titles in one document. Hooks are provided so that additional titling elements can be defined and printed by \maketitle.

The \thanks command is enhanced to provide various configurations for both the marker symbol and the layout of the thanks notes.

2.1   Commands and environments

2.1.1   Restyling the titling

The titling package provides a configurable \maketitle command.

\pretitle
\posttitle
\preauthor
\postauthor
\predate
\postdate

These six commands each have a single argument, like \pretitle{⟨*text*⟩}and \posttitle{⟨*text*⟩}. The argument to these commands control the typesetting of the standard elements of the document's \maketitle command. The \title is effectively processed between the \pretitle and \posttitle commands; that is, like:

{\pretitle \title \posttitle}

and similarly for the \author and \date commands. The commands are initialised to mimic the normal result of \maketitle typesetting in the article/report classes. For example, the default definitions of the \...title... and \...author... commands are:

```
\pretitle{\begin{center}\LARGE}
\posttitle{\par\end{center}\vskip 0.5em}
\preauthor{\begin{center}
            \large \lineskip 0.5em%
            \begin{tabular}[t]{c}}
\postauthor{\end{tabular}\par\end{center}}
\predate{\begin{center}\large}
\postdate{\par\end{center}}
```

They can be changed to obtain different effects. For example to get a right justified sans-serif title and a left justifed small caps date:

```
\pretitle{\begin{flushright}\LARGE\sffamily}
\posttitle{\par\end{flushright}\vskip 0.5em}
\predate{\begin{flushleft}\large\scshape}
\postdate{\par\end{flushleft}}
```

NOTE: The above commands differ from the similar ones in version 1.0 of this package. So, if you have upgraded from v1.0 to v2.0, beware.

As another example, the following (slightly edited) was sent to me by Jason Harrison (`harrison@cs.ubc.ca`) on 2002/02/18.

> I'm trying to format my latest conference paper for a conference that doesn't supply a latex class (only a MSWord example, sigh) and I ran into a problem that I'm not sure if it's an interaction of the titling package [it's not] or just no one ever cares because hitting return gets you an output when latex complains:

```
! Argument of \reserved@a has an extra }.
<inserted text>
                \par
l.46         \end{tabular}}
? x
```

> What I did was to define my authors list using the following methodology:

```
\preauthor{}
\DeclareRobustCommand{\authorthing}{
\begin{center}
\begin{tabular}{ccc}
  Jason Harrison & Kellogg S. Booth & Brian D. Fisher \\
  \multicolumn{3}{c}{Department of Computer Science} \\
  \multicolumn{3}{c}{and} \\
  \multicolumn{3}{c}{Media and Graphics Interdisciplinary Centre}\\
  \multicolumn{3}{c}{University of British Columbia, Vancouver, BC Canada}
          \end{tabular}
\end{center}}
\author{\authorthing}
\postauthor{}
```

> Thus latex does not complain and I get the formatting I want. Perhaps you'd add this to the titling documentation for those of us who can't follow the usual conventions because of the strange conventions of a strange conference.

Changing the above example a little, when authors are from different organisations you may be required to flag these. This is one way of doing that.

```
\preauthor{\large}
\DeclareRobustCommand{\authoring}{
  \begin{center}
    Smith\textsuperscript{\dag} and Wesson\textsuperscript{\#} \\
    \textsuperscript{\dag}House, \textsuperscript{\#}Barn       \\
    The OK Corral
  \end{center}}
```

```
\author{\authoring}
\postauthor{\par}
```

\droptitle   The `\maketitle` command puts the title at a particular height on the page. Unless it is used with the titlepage option of the standard classes it starts a new page. You can change the vertical position of the title via the length `\droptitle`. Giving this a positive value will lower the title and a negative value will raise it. The default definition is:

`\setlength{\droptitle}{0pt}`.

\maketitlehooka
\maketitlehookb
\maketitlehookc
\maketitlehookd
      The `\maketitle` command as defined by the titling package is essentially

```
\renewcommand{\maketitle}{%
  \vspace*{\droptitle}
  \maketitlehooka
  {\pretitle \title \posttitle}
  \maketitlehookb
  {\preauthor \author \postauthor}
  \maketitlehookc
  {\predate \date \postdate}
  \maketitlehookd
}
```

The four hook commands are provided so that additional elements may be added to `\maketitle`. These are initially defined to do nothing but can be renewed. For example, some publications want a statement about where an article is published immediately before the actual titling text. The following defines a command `\published` that can be used to hold the publishing information which will then be automatically printed by `\maketitle`.

```
\newcommand{\published}[1]{%
   \gdef\puB{#1}}
\newcommand{\puB}{}
\renewcommand{\maketitlehooka}{%
   \par\noindent \puB}
```

You can then say:

```
\published{Originally published in
         \textit{The Journal of ...}\thanks{Reprinted with permission}}
...
\maketitle
```

to print both the published and the normal titling information. Note that nothing extra had to be done in order to use the `\thanks` command in the argument to the new `\published` command.

titlingpage   When one of the standard classes is used with the titlepage option, `\maketitle`

puts the title elements on an unnumbered page and then starts a new page numbered page 1. The standard classes also provide a `titlepage` environment which starts a new unnumbered page and at the end starts a new page numbered 1. You are entirely responsible for specify exactly what and where is to go on this title page. Do not use `\maketitle` within the `titlepage` environment as it will start yet another page.

The `titlingpage` environment falls between the `titlepage` option and the `titlepage` environment. Within the `titlingpage` environment you can use the `\maketitle` command, and any others you wish. The pagestyle is `empty`; at the end it starts another ordinary page numbered page 1.

For example, to put both the title and an abstract on a title page

```
\begin{document}
\begin{titlingpage}
\setlength{\droptitle}{30pt} % lower the title
\maketitle
\begin{abstract}...\end{abstract}
\end{titlingpage}
```

By default, titling information is centered with respect to the width of the text block. Occasionally someone asks on the `comp.text.tex` newsgroup how to center the titling information on a title page with respect to the width of the physical page. If the textblock is centered with respect to the physical page, then the default centering suffices. If the textblock is not physically centered, then the titling information either has to be shifted horizontally or `\maketitle` has to be made to think that the textblock has been shifted horizontally.

One solution is to use the `titlingpage` environment and the `chngpage` package[1], which lets you temporarily adjust the location of the textblock. Remember that TeX starts its page measurements at a distance of 1 inch + `\hoffset` (default 0pt) in from the left side of the physical sheet; this is the zero point for the `\evensidemargin` and the `\oddsidemargin`. Let H be the value of (1 inch + `\hoffset`), M be the `\...sidemargin`, T be the `\textwidth` and P be the width of the physical page, and let X be the distance between the righthand edge of the text block and the right side of the physical sheet. Then

$$H + M + T + X = P$$

and the textblock is $H + M$ in from the left side of the page. For the textblock to be centered on the physical page, we must have $X = H + M$. If the textblock is not physically centered, then the margin value, M can be adjusted to made it so. Let d be the amount to be added to M to center the textblock. Some simple algebra leads to

$$d = (P - T)/2 - M - H$$

as the amount to be added to the `\...sidemargin` to center the textblock. For example, assume that the `\textwidth` is 5 inches, the `\...sidemargin`s are both

---

[1] Available from CTAN as file `chngpage.sty`.

2 inches, and \hoffset is zero (i.e., H is 1 inch). Then to center the textblock, 1.25 inches must be subtracted from the margin if it is printed on US letterpaper, where the paper width is 8.5 inches. A different adjustment would be required if A4 paper was used instead. See the chngpage documentation for details, but a physically centred title can now be achieved by

```
\usepackage{chngpage}
...
\begin{titlingpage}
   \chngtext{}{}{-1.25in}{-1.25in}{}
   \maketitle
\end{titlingpage}
```

\calccentering   If you don't want to do the above calculations by hand, \calccentering{⟨*length*⟩}{⟨*margin*⟩} will do it for you. The ⟨*length*⟩ argument must be the name of a pre-existing length command, including the backslash, and the ⟨*margin*⟩ is the value of the appropriate \evensidemargin or \oddsidemargin. After calling \calccentering, ⟨*length*⟩ is the amount to be added to the margin to center the textblock. An example usage is

```
\begin{titlingpage}
   \newlength{\addtoeven}
   \newlength{\addtoodd}
   \calccentering{\addtoeven}{\evensidemargin}
   \calccentering{\addtoodd}{\oddsidemargin}
   \chngtext{}{}{\addtoeven}{\addtoodd}{}
   ....
\end{titlingpage}
```

The \calccentering macro requires the length \paperwidth to be set to the physical width of the paper you will be printing on. If your paper matches the paper size option in for the document class then this is done for you.

You do not necessarily have to define a new length for the purposes of \calccentering. Any existing length will do, such as \unitlength, provided it will be otherwise unused between performing the calculation and changing the margins (and that you can, if necessary reset it to its original value — the default value for \unitlength is 1pt). There are also some lengths provided by the package that you may consider using, such as \droptitle, \thanksmarkwidth or \thanksmargin.

\thetitle   In the usual document classes, the \title, \author and \date macros used
\theauthor   for \maketitle are unavailable once \maketitle has been issued. The package
\thedate   provides the \thetitle, \theauthor and \thedate commands that can be used for printing these elements of the title later in the document, if desired.

The titling package prevents the standard classes' habit of automatically killing the titling commands after \maketitle has been issued. You can have multiple \title, \author, \date and \maketitle commands in your document if you wish. For example, some reports are issued with a title page, followed by an executive

summary, and then they have another, possibly modified, title at the start of the main body of the report. This can be accomplished like this:

```
\title{Cover title}
...
\begin{titlingpage}
\maketitle
\end{titlingpage}
...
\title{Body title}
\maketitle
...
```

\killtitle The \killtitle macro makes all aspects of titling, including \thetitle etc,
\keepthetitle unavailable from the point that it is issued (using this command will save some
\emptythanks macro space if the \thetitle, etc., commands are not required). Using this command is the titling package's manual version of the automatic killing performed by the standard classes. The \keepthetitle command performs a similar function, except that it keeps the \thetitle, \theauthor and \thedate commands, while killing everything else. The \emptythanks command discards any text from prior use of \thanks. This command is useful when \maketitle is used multiple times — the \thanks commands in each use just stack up the texts for output at each use, so each subsequent use of \maketitle will output all previous \thanks texts together with any new ones. To avoid this, put \emptythanks before each \maketitle after the first.

2.1.2 Restyling the thanks

The titling package provides a configurable \thanks command. As a side effect of modifications to the \thanks command, the thanks text can be one or more paragraphs, unlike the standard \thanks text which is restricted to a single paragraph.

\thanksmarkseries In the standard classes, \thanks{} are marked with symbols in the titling
\symbolthanksmark and footnotes unless on a titlepage where they are marked with numbers. The command \thanksmarkseries{⟨format⟩} command can be used to change the marking style. The ⟨format⟩ argument is the name of one of the formats for printing a counter. The name is the same as that of a counter format but without the backslash. The default matches the standard class defaults. To have the titlepage marks as symbols instead of numerals do:
\thanksmarkseries{fnsymbol}
or to use lowercase letters do:
\thanksmarkseries{alph}
Just for convenience the \symbolthanksmark command sets the series to be footnote symbols (so you don't have to use the first example above). Using the standard classes the potential names for ⟨format⟩ are: arabic, roman, Roman, alph, Alph, and fnsymbol. Other classes or packages may provide more or fewer names.

\continuousmarks The \thanks command uses the footnote counter, and normally the counter

is zeroed after the titling so that the footnote marks start from 1. If the counter should not be zeroed, then just specify `\continuousmarks`. This might be required if numerals are used as the thanks markers.

`\thanksheadextra` The `\thanksheadextra{`⟨*pre*⟩`}{`⟨*post*⟩`}` command will insert ⟨*pre*⟩ and ⟨*post*⟩
`\thanksfootextra` before and after the thanks markers in the titling block.

Similarly `\thanksfootextra{`⟨*pre*⟩`}{`⟨*post*⟩`}` inserts ⟨*pre*⟩ and ⟨*post*⟩ before and after the thanks markers in the footnotes.

By default ⟨*pre*⟩ and ⟨*post*⟩ are empty in both cases. For example, to put parentheses round the titling markers and to have a full stop (a period) after the footnote markers do:

`\thanksheadextra{(}{)}` and
`\thanksfootextra{}{.}`

`\thanksmark` It is sometimes desireable to have the same thanks text be applied to, say, four out of six authors, these being the first 3 and the last one. The command `\thanksmark{`⟨*n*⟩`}` is similar to `\footnotemark[`⟨*n*⟩`]` in that it prints a thanks mark identical to that of the ⟨*n*⟩'th `\thanks` command. No changes are made to any thanks in the footnotes.

`\thanksgap` The marks in the titling block printed by the `\thanks` and `\thanksmark` commands take zero space. This is acceptable if they come at the end of a line, but not in the middle of a line. Use the `\thanksgap{`⟨*length*⟩`}` command immediately after a mid-line `\thanks` or `\thanksmark` command to add ⟨*length*⟩ amount of space before the next word. For example, if there are three authors from two different institutions:

```
\author{Alpha\thanks{ABC},
        Omega\thanks{XYZ}\thanksgap{1ex} and
        Beta\thanksmark{1}}
```

`\thanksmarkwidth` The thanks mark in the footnote is typeset right justified in a box of width `\thanksmarkwidth`. The first line of the thanks text starts after this box. The initialisation is `\setlength{\thanksmarkwidth}{1.8em}` giving the default position.

`\thanksmargin` The value of the length `\thanksmargin` controls the indentation the second and subsequent lines of the thanks text, with respect to the end of the mark box. As examples:

`\setlength{\thanksmargin}{0em}` will align the left hand ends of of a multiline thanks text, while:

`\setlength{\thanksmargin}{-\thanksmarkwidth}` will left justify any second and subsequent lines of the thanks text. This last setting is the initialised value, giving the default appearance.

`\thanksfootmark` A thanks mark in the footnote region is typeset by `\thanksfootmark`. The code for this is roughly:

```
\newcommand{\thanksfootmark}{%
  \hbox to\thanksmarkwidth{\hfil\normalfont%
    \thanksscript{\thanksfootpre \thefootnote \thanksfootpost}}}
```

where `\thanksfootpre` and `\thanksfootpost` are specified via the `\thanksfootextra`

macro. You should not need to change the definition of `\thanksfootmark` but you may want to change the default definitions of one or more of the macros it uses.

`\thanksscript`   Note that the thanks mark, together with the `\...pre` and `\...post` commands form the argument to the `\thanksscript` command. This is initially defined as:

`\newcommand{\thanksscript}[1]{\textsuperscript{#1}}`

so that `\thanksscript` typesets its argument as a superscript, which is the default for thanks notes. If you would prefer the mark to be set at the baseline instead, for example, just do:

`\renewcommand{\thanksscript}[1]{#1}`, and if you also wanted very small symbols on the baseline you could do:

`\renewcommand{\thanksscript}[1]{\tiny #1}`. Alternatively

`\renewcommand{\thanksscript}[1]{#1}`
`\thanksfootextra{\bfseries}{.}`

will give a bold baseline mark followed by a dot.

Using combinations of `\thanksscript`, `\thanksmargin`, `\thanksmarkwidth` and `\thanksfootextra` you can easily define different layouts for the thanks footnotes. Here are some sample combinations, but to shorten them I have ignored any `\renewcommand`s and `\setlength`s, leaving these to be implied as necessary.

- Setting `\thanksfootextra{}{\hfill}` left justifies the mark in its box:

  - `\thanksscript{\llap{#1\space}}`, `\thanksmarkwidth{0em}` and `\thanksmargin{0em}` puts the baseline mark in the margin and the text left justified.

  - Using `\thanksscript{#1}`, `\thanksmarkwidth{1em}` and `\thanksmargin{-\thanksmarkwidth}` makes the baseline mark and text left adjusted.

  - `\thanksscript{#1}`, `\thanksmarkwidth{1em}` and `\thanksmargin{0em}` puts the baseline mark left adjusted and the text indented and aligned, like this marked sentence is typeset.

- Setting `\thanksfootextra{ }` and `\thanksscript{#1}` right justifies the baseline mark and a space in the mark box:

  - The normal style is defined by `\thanksmarkwidth{1.8em}` and `\thanksmargin{-\thanksmarkwidth}` which put the mark indented and the text left adjusted, like a normal indented paragraph.

  - `\thanksmarkwidth{1.8em}` and `\thanksmargin{0em}` put the mark indented and the text indented and aligned.

`\makethanksmark`   The macro `\makethanksmark` typesets both the thanks marker (via `\thanksfootmark`)
`\makethanksmarkhook` and the thanks text. You probably will not need to change its default definition. Just in case, though, `\makethanksmark` calls the macro `\makethanksmarkhook` before it does any typesetting. The default definition of this is:
`\newcommand{\makethanksmarkhook}{}`, which does nothing.

9

You can redefine `\makethanksmarkhook` to do something useful. For example, if you wanted a slightly bigger baseline skip you could do:

`\renewcommand{\makethanksmarkhook}{\fontsize{8}{11}\selectfont}`

where the numbers 8 and 11 specify the point size of the font and the baseline skip respectively. In this example 8pt is the normal `\footnotesize` in a 10pt document, and 11pt is the baselineskip for `\footnotesize` text in an 11pt document (the baseline skip is 9.5pt in a 10pt document); adjust these numbers to suit.

`\thanksrule` By default, there is no rule above `\thanks` text that appears in a `titlingpage`
`\usethanksrule` environment. If you want a rule in that environment, put `\usethanksrule` be-
`\cancelthanksrule` fore the `\maketitle` command, which will then print a rule according to the current definition of `\thanksrule`. `\thanksrule` is initialised to be a copy of `\footnoterule` as it is defined at the end of the preamble. The definition of `\thanksrule` can be changed after `\begin{document}`. If the definition of `\thanksrule` is modified and a `\usethanksrule` command has been issued, then the redefined rule may also be used for footnotes. Issuing the command `\cancelthanksrule` will cause the normal `\footnoterule` definition to be used from thereon; another `\usethanksrule` command can be issued later if you want to swap back again.

The parameters for the vertical positioning of footnotes and thanks notes, and the default `\footnoterule` are described below. You may need to change one or more of these if the vertical spacings of footnotes and thanks notes are meant to be different.

`\footnotesep` The length `\footnotesep` controls the vertical spacing between footnotes
`\skip` (and thanks notes), and is initialised by the class to give no extra spacing for a `\footnotesize` font. You can change the spacing by

`\addtolength{\footnotesep}{...}`

The length `\skip\footins` defines the vertical spacing between the bottom of the main text and the top of the first line of the first footnote. Likewise this can be changed by

`\addtolength{\skip\footins}{...}`

The total vertical distance between the bottom of the main text and the baseline of the first line of the first footnote is `\footnotesep + \skip\footins`.

`\footnoterule` The `\footnoterule` command is defined in the LaTeX kernel and redefined in the standard classes. An '@less' definition, which is slightly less efficient than that in the classes, is:

```
\renewcommand{\footnoterule}{%
   \kern -3pt  % call this kerna
   \hrule height 0.4pt width 0.4\columnwidth
   \kern 2.6pt % call this kernb
}
```

This produces a horizontal rule from the left margin to 40% of the `\columnwidth` with thickness 0.4pt. The rule must not take up any vertical space, which means that `kerna + kernb` must equal the thickness of the rule. The rule is located a distance `\skip\footins + kerna` below the bottom of the main text. So, to

move the rule upwards, decrease the value of `kerna` and increase that of `kernb`, and the reverse to move the rule downwards.

`\appendiargdef` This is only noted here as someone may find it useful in another context. Get the patchcmd package [Dow00] if you need to make other additions to definitions. The internal command `\appendiargdef{⟨macro⟩}{⟨stuff⟩}` appends ⟨stuff⟩ at the end of the current definition of ⟨macro⟩, where ⟨macro⟩ is the name of a macro (including the backslash) which takes a single argument. For example the following are two equivalent definitions of `\mymacro`:

```
\newcommand{\mymacro}[1]{#1 is a violinist}
\appendiargdef{\mymacro}{ in spite of being tone deaf}
% or
\newcommand{\mymacro}[1]{# is a violinist in spite of being tone deaf}
```

The usual advice about creating a one-column `abstract` in a two-column document is to write code like this:

```
\documentclass[twocolumn...]{...}
...
\twocolumn[
   \begin{@twocolumnfalse}
     \maketitle                % need full-width title
     \begin{abstract}
        abstract text...
     \end{abstract}
   \end{@twocolumnfalse}
]
... hand make footnotes for any \thanks commands
...
```

The titling package gives no assistance for doing this, but the abstract package [Wil01] provides a command that caters for the somewhat non-intuitive `...  hand make footnotes ...` aspect.

3   The package code

To try and avoid name clashes, all the internal commands include the string `@bs` (this code was initially part of the abstract package).

3.1   Preliminaries

Announce the name and version of the package, which requires LaTeX 2_ε.

```
1 ⟨∗usc⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{titling}[2004/09/26 v2.1c maketitle typesetting]
4
```

11

The original code for \maketitle is shown in Section A. Changes to \maketitle and friends are defined here.

To provide some flexibilty in the titling style of the document, user level commands are provided that can be changed to reconfigure the appearance resulting from \maketitle.

\pretitle
\@bspretitle
\posttitle
\@bsposttitle
\preauthor
\@bspreauthor
\postauthor
\@bspostauthor
\predate
\@bspredate
\postdate
\@bspostdate

```
5 \newcommand{\pretitle}[1]{\def\@bspretitle{#1}}
6 \newcommand{\posttitle}[1]{\def\@bsposttitle{#1}}
7 \newcommand{\preauthor}[1]{\def\@bspreauthor{#1}}
8 \newcommand{\postauthor}[1]{\def\@bspostauthor{#1}}
9 \newcommand{\predate}[1]{\def\@bspredate{#1}}
10 \newcommand{\postdate}[1]{\def\@bspostdate{#1}}
11
```

These are defined initially to mimic the normal LaTeX style.

```
12 \if@titlepage
13   \pretitle{\begin{center}\LARGE}
14   \posttitle{\par\end{center}\vskip 2em}
15   \preauthor{\begin{center}
16     \large \lineskip .75em%
17     \begin{tabular}[t]{c}}
18   \postauthor{\end{tabular}\par\end{center}}
19   \predate{\begin{center}\large}
20   \postdate{\par\end{center}}
21 \else
22   \pretitle{\begin{center}\LARGE}
23   \posttitle{\par\end{center}\vskip 0.5em}
24   \preauthor{\begin{center}
25     \large \lineskip .5em%
26     \begin{tabular}[t]{c}}
27   \postauthor{\end{tabular}\par\end{center}}
28   \predate{\begin{center}\large}
29   \postdate{\par\end{center}}
30 \fi
31
```

\maketitlehooka
\maketitlehookb
\maketitlehookc
\maketitlehookd

The four hooks which will be called by \maketitle. These are initially vacuous.

```
32 \newcommand{\maketitlehooka}{}
33 \newcommand{\maketitlehookb}{}
34 \newcommand{\maketitlehookc}{}
35 \newcommand{\maketitlehookd}{}
36
```

\thanksmarkseries
\@bsmarkseries
\symbolthanksmark

These are for specifying the kind of series for thanks markers.

```
37 \newcommand{\thanksmarkseries}[1]{%
38   \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}}
39 \newcommand{\symbolthanksmark}{\thanksmarkseries{\fnsymbol}}
```

12

\continuousmarks
\@bscontmark

These are for (non) zeroing of the footnote counter.

```
40 \newcommand{\@bscontmark}{\setcounter{footnote}{0}}
41 \newcommand{\continuousmarks}{\def\@bscontmark{}}
```

\thanksheadextra
\@bsthanksheadpre
\@bsthanksheadpost

These are for inserting stuff before and after a mark in the titling.

```
42 \newcommand{\thanksheadextra}[2]{%
43   \def\@bsthanksheadpre{#1}%
44   \def\@bsthanksheadpost{#2}}
```

\thanksfootextra
\thanksfootpre
\thanksfootpost

These are for inserting stuff before and after a thanks mark in the footnoting.

```
45 \newcommand{\thanksfootextra}[2]{%
46   \def\thanksfootpre{#1}%
47   \def\thanksfootpost{#2}}
48
```

\thanksmark

This adds a thanks mark. The \footnotemark could have been used directly but it is fragile in a moving argument.

```
49 \DeclareRobustCommand{\thanksmark}[1]{\footnotemark[#1]}
```

\thanksgap

This specifies some horizontal space.

```
50 \newcommand{\thanksgap}[1]{\hspace{#1}}
```

\tamark

This stores the current definition of \@thefnmark. For some reason using \@thefnmark directly only gave the last value.

```
51 \newcommand{\tamark}{\@thefnmark}
52
```

\thanksmarkwidth

A length determining the size of the box for typesetting a thanks marker.

```
53 \newlength{\thanksmarkwidth}
```

\thanksmargin

A length determining the inset of thanks footnotes.

```
54 \newlength{\thanksmargin}
```

\thanksscript

A wrapper round the actual mark stuff to be typeset. This is initialised to give a superscript mark.

```
55 \newcommand{\thanksscript}[1]{\textsuperscript{#1}}
```

\makethanksmarkhook

A vacuous macro used as a hook into \makethanksmark.

```
56 \newcommand{\makethanksmarkhook}{}
```

\thanksfootmark

This typesets the thanks footnote mark.

```
57
58 \newcommand{\thanksfootmark}{%
59   \hb@xt@\thanksmarkwidth{\hfil\normalfont\thanksscript{%
60     \thanksfootpre \tamark \thanksfootpost}}}
61
```

13

\makethanksmark   This sets the general indentations for the thanks footnote, and typesets the mark. The code is a simplified version of that for typeseeting ToC entries.

```
62 \newcommand{\makethanksmark}{%
63    \leavevmode%
64    \makethanksmarkhook\relax
65    \parindent 1em\noindent
66    \leftskip\thanksmargin\relax
67    \advance\leftskip \thanksmarkwidth \null\nobreak\hskip -\leftskip
68    \thanksfootmark
69 }
70
```

\usethanksrule   Simple macros that let \footnoterule to another rule definition.
\cancelhanksrule

```
71 \newcommand{\usethanksrule}{\let\footnoterule\thanksrule}
72 \newcommand{\cancelthanksrule}{\let\footnoterule\@bsfootnoterule}
73
```

     Now set up the thanks defaults.

```
74 \if@titlepage
75    \thanksmarkseries{arabic}     % normal footnote numbers
76 \else
77    \thanksmarkseries{fnsymbol}
78 \fi
79 \thanksheadextra{}{}
80 \thanksfootextra{}{}
81 \setlength{\thanksmarkwidth}{1.8em}
82 \setlength{\thanksmargin}{-\thanksmarkwidth}
83
```

\thanksrule   These are saved versions of the \footnoterule definition as it is at the end of the
\@bsfootnoterule preamble.

```
84 \AtBeginDocument{%
85    \let\thanksrule\footnoterule
86    \let\@bsfootnoterule\footnoterule
87 }
88
```

\droptitle   A titling block has \droptitle amount of additional vertical space above it (normally zero).

```
89 \newlength{\droptitle}
90 \setlength{\droptitle}{0pt}
91
```

\maketitle   The following is a modification of \maketitle as in the article, report, and book classes (see Section A for the original definition).

```
92 \providecommand{\maketitle}{}
93 \if@titlepage
94    \renewcommand{\maketitle}{\begin{titlepage}%
95       \let\footnotesize\small
```

```
 96      \let\footnoterule\relax
 97      \let \footnote \thanks
 98      \@bsmarkseries
 99       \def\@makefnmark{\rlap{\@textsuperscript{%
100         \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
101       \long\def\@makefntext##1{\makethanksmark ##1}
102      \null\vfil
103      \vskip 60\p@
104      \vspace*{\droptitle}
105      \maketitlehooka
106      {\@bspretitle \@title \@bsposttitle}
107      \maketitlehookb
108      {\@bspreauthor \@author \@bspostauthor}
109      \maketitlehookc
110      {\@bspredate \@date \@bspostdate}
111      \maketitlehookd
112      \par
113      \@thanks
114      \vfil\null
115      \end{titlepage}%
116      \@bscontmark  %  \setcounter{footnote}{0}%
117 %%%     \@bsmtitlempty
118    } % end titlepage defs
119 \else
120   \renewcommand{\maketitle}{\par
121     \begingroup
122       \@bsmarkseries
123       \def\@makefnmark{\rlap{\@textsuperscript{%
124         \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
125       \long\def\@makefntext##1{\makethanksmark ##1}
126      \if@twocolumn
127        \ifnum \col@number=\@ne
128          \@maketitle
129        \else
130          \twocolumn[\@maketitle]%
131        \fi
132      \else
133        \newpage
134        \global\@topnum\z@
135        \@maketitle
136      \fi
137      \thispagestyle{plain}\@thanks
138     \endgroup
139     \@bscontmark  %  \setcounter{footnote}{0}%
140 %%%     \@bsmtitlempty
141    } % end non-titlepage
```

\@maketitle   I use \def\@maketitle to account for the cases where the main class does not
have titling commands, and to ensure an existing \@maketitle gets overridden.

```
142
```

```
143  \def\@maketitle{%
144    \newpage
145    \null
146    \vskip 2em%
147          \vspace*{\droptitle}
148    \maketitlehooka
149    {\@bspretitle \@title \@bsposttitle}
150    \maketitlehookb
151    {\@bspreauthor \@author \@bspostauthor}
152    \maketitlehookc
153    {\@bspredate \@date \@bspostdate}
154    \maketitlehookd
155    \par
156    \vskip 1.5em}
157 \fi
158
```

titlingpage The `titlingpage` environment sets the pagestyle to be empty, disables the footnote rule and ensures that the page is single column. At the end it switches back to twocolumn if neccessary, and then starts a new page as number 1.

```
159 \newenvironment{titlingpage}%
160   {%
161    \let\footnoterule\relax
162    \let\footnotesize\small
163    \if@twocolumn
164      \@restonecoltrue\onecolumn
165    \else
166      \@restonecolfalse
167    \fi
168    \thispagestyle{empty}
169    \setcounter{page}\@ne
170   }{%
171    \thispagestyle{empty}
172    \if@restonecol\twocolumn \else \newpage \fi
173    \if@twoside\else \setcounter{page}\@ne\fi}
174
```

\calccentering This macro calculates the amount to be added to a `...sidemargin` in order to center the textblock. Call as
`\calccentering{\length}{...sidemargin}` and it sets `\length` to the required value.

```
175 \newcommand{\calccentering}[2]{
176   #1 = \paperwidth
177   \advance #1 by -\textwidth
178   \divide #1 by \tw@
179   \advance #1 by -#2
180   \advance #1 by -\hoffset
181   \advance #1 by -1in
182 }
```

\emptythanks   This macro discards all prior \thanks texts.

184 `\newcommand{\emptythanks}{\global\let\@thanks\@empty}`
185

\@bsmtitlempty   `\@bsmtitlempty` is a helper macro to save some macro space. It empties some elements of `\maketitle`.

```
186 \newcommand{\@bsmtitlempty}{%
187   \global\let\maketitle\relax
188   \global\let\@maketitle\relax
189   \global\let\title\relax
190   \global\let\author\relax
191   \global\let\date\relax
192   \global\let\thanksmarkseries\relax
193   \global\let\thanksheadextra\relax
194   \global\let\thanksfootextra\relax
195   \global\let\thanksmark\relax
196   \global\let\thanksgap\relax
197 }
198
```

\keepthetitle   This macro undefines all the titling commands except for \thetitle, \theauthor and \thedate.

```
199 \newcommand{\keepthetitle}{%
200   \@bsmtitlempty
201   \global\let\thanks\relax
202   \global\let\and\relax
203   \global\let\@thanks\@empty
204   \global\let\@title\@empty
205   \global\let\@author\@empty
206   \global\let\@date\@empty
207 }
208
```

\killtitle   \killtitle undefines the remaining macros of \maketitle.

```
209 \providecommand{\killtitle}{}
210 \renewcommand{\killtitle}{%
211   \keepthetitle
212   \global\let\thetitle\relax
213   \global\let\theauthor\relax
214   \global\let\thedate\relax
215 }
216
```

\appendiargdef   The code is copied from the abstract package.

```
217 \providecommand{\appendiargdef}[2]{\begingroup
218   \toks@\expandafter{#1{##1}#2}%
219   \edef\@bsx{\endgroup \def\noexpand#1####1{\the\toks@}}%
```

```
220     \@bsx}
221
```

\thetitle
\theauthor
\thedate

In order to make the `\title`, etc., values available for printing their definitions need extending to save their arguments (see Section A for their original definitions). We have to make sure that extraneous material, like `\thanks`, is excluded from the saved texts.

```
222 \appendiargdef{\title}{%
223   \begingroup
224     \renewcommand{\thanks}[1]{}
225     \renewcommand{\thanksmark}[1]{}
226     \renewcommand{\thanksgap}[1]{}
227     \protected@xdef\thetitle{#1}
228   \endgroup}
229 \appendiargdef{\author}{%
230   \begingroup
231     \renewcommand{\thanks}[1]{}
232     \renewcommand{\and}{\unskip, }
233     \renewcommand{\thanksmark}[1]{}
234     \renewcommand{\thanksgap}[1]{}
235     \protected@xdef\theauthor{#1}
236   \endgroup}
237 \appendiargdef{\date}{%
238   \begingroup
239     \renewcommand{\thanks}[1]{}
240     \renewcommand{\thanksmark}[1]{}
241     \renewcommand{\thanksgap}[1]{}
242     \protected@xdef\thedate{#1}
243   \endgroup}
244
```

The end of this package.

```
245 ⟨/usc⟩
```

A   Original code for making a title

These are the kernel commands (from `ltsect.dtx`) for the elements of `\maketitle`.

```
\def\title#1{\gdef\@title{#1}}
  \def\@title{\@latex@error{No \noexpand\title given}\@ehc}
\def\author#1{\gdef\@author{#1}}
  \def\@author{\@latex@warning@no@line{No \noexpand\author given}}
\def\date#1{\gdef\@date{#1}}
  \def\@date{\today}
\def\thanks#1{\footnotemark
  \protected@xdef\@thanks{\@thanks
    \protect\footnotetext[\the\c@footnote]{#1}}%
}
```

18

```
\let\@thanks\@empty
\def\and{%                      % \begin{tabular}
  \end{tabular}%
  \hskip 1em \@plus.17fil%
  \begin{tabular}[t]{c}}%   % \end{tabular}
```

This is the version of \maketitle and \@maketitle from classes.dtx.

```
\if@titlepage
\newcommand{\maketitle}{\begin{titlepage}%
  \let\footnotesize\small
  \let\footnoterule\relax
  \let \footnote \thanks
  \null\vfil
  \vskip 60\p@
  \begin{center}%
    {\LARGE \@title \par}%
    \vskip 3em%
    {\large
     \lineskip .75em%
     \begin{tabular}[t]{c}%
       \@author
     \end{tabular}\par}
     \vskip 1.5em%
    {\large \@date \par}%
  \end{center}\par
  \@thanks
  \vfil\null
  \end{titlepage}%
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@maketitle\relax
  \global\let\@thanks\@empty
  \global\let\@author\@empty
  \global\let\@date\@empty
  \global\let\@title\@empty
  \global\let\title\relax
  \global\let\author\relax
  \global\let\date\relax
  \global\let\and\relax
}
\else
\newcommand{\maketitle}{\par
  \begingroup
    \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
    \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
    \long\def\@makefntext##1{\parindent 1em\noindent
      \hb@xt@1.8em{%
```

```
              \hss\@textsuperscript{\normalfont\@thefnmark}}##1}%
        \if@twocolumn
          \ifnum \col@number=\@ne
            \@maketitle
          \else
            \twocolumn[\@maketitle]%
          \fi
        \else
          \newpage
          \global\@topnum\z@
          \@maketitle
        \fi
        \thispagestyle{plain}\@thanks
      \endgroup
      \setcounter{footnote}{0}%
      \global\let\thanks\relax
      \global\let\maketitle\relax
      \global\let\@maketitle\relax
      \global\let\@thanks\@empty
      \global\let\@author\@empty
      \global\let\@date\@empty
      \global\let\@title\@empty
      \global\let\title\relax
      \global\let\author\relax
      \global\let\date\relax
      \global\let\and\relax
    }

    \def\@maketitle{%
      \newpage
      \null
      \vskip 2em%
      \begin{center}%
        \let \footnote \thanks
        {\LARGE \@title \par}%
        \vskip 1.5em%
        {\large
         \lineskip .5em%
         \begin{tabular}[t]{c}%
           \@author
         \end{tabular}\par}
         \vskip 1em%
        {\large \@date}%
      \end{center}
      \par
      \vskip 1.5em}
    \fi
```

References

[Dow00] Michael J. Downes. *The patchcmd package.* July, 2000. (Available from CTAN in subdirectory `/patchcmd`)

[GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley Publishing Company, 1994.

[Wil01] Peter Wilson. *The abstract package.* February, 2001. (Available from CTAN in subdirectory `/abstract`)

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.