Non-Floating Margin Notes with \texttt{marginnote} Package\textsuperscript{*}

Markus Kohm\textsuperscript{†}

2009/05/06

Abstract

In \texttt{LaTeX} the command \texttt{\textbackslash marginpar\{\textit{left}\}\{\textit{right}\}} might be used to create a note in the margin. But there is a problem with this command: it creates a special kind of float. For this it cannot be used e.g. at floats or footnotes. Package \texttt{marginnote} supports another command \texttt{\textbackslash marginnote} to create notes in the margin. This does not use a kind of float and for this does not have the disadvantage of \texttt{\textbackslash marginpar}. But there might be other problems…

1 How to Use \texttt{marginnote} Package

First of all you have to load. You may use:

\texttt{\usepackage{marginnote}}

to do so.

The command \texttt{\textbackslash marginnote\{\textit{left}\}\{\textit{right}\}\{\textit{voffset}\}} may be used to set a margin note using \texttt{marginnote}. The first optional argument and the mandatory argument are same using \texttt{\textbackslash marginpar} from the \texttt{LaTeX} kernel. Even \texttt{\textbackslash reversemarginpar} will be considered. The note \texttt{\textit{left}} or \texttt{\textit{right}} will be put at the current vertical position. Second optional argument \texttt{\textit{voffset}} may be used to adjust the vertical position of the margin note. Use a negative dimension to move it up or a positive dimension to move it down.

At some environments, e.g. \texttt{framed} from the \texttt{framed} package the horizontal placement of the margin notes are not correct. In this case you may redefine \texttt{\textbackslash marginnotopleftadjust} and \texttt{\textbackslash marginnoterightadjust} to fix this. Note that these are macros not lengths! So you have to use \texttt{\renewcommand}, \texttt{\def} or \texttt{\let} to change them. You may e.g. use

\texttt{\begingroup}

\textsuperscript{*}This file has version number v1.1d, last revised 2009/05/06.
\textsuperscript{†}Email: komascript@gmx.info
at your preamble after loading package `framed` to fix the problem using `framed` environment.

NOTE: \marginnoteleftadjust and \marginnoterightadjust will be used only, if the correct horizontal position cannot be determined using PDF\TeX\ features (\pdfsavepos and \pdflastxpos). So if you are using PDF\TeX\ with PDF output or PDF\TeX\ with PDF\TeX\-version since 1.40 or \Xe\TeX\ you will not need to use the example code above, but you will need at least two \TeX\ runs to get the correct horizontal positions of the margin notes.

Package `marginnote` needs to know the real width of the type area to find the right margin. While some environments (e.g. those of package `framed`) change \textwidth, `marginnote` defines its own text width macro. If you change type area after \begin{document} you should add

```
edef\marginnotetextwidth{\the\textwidth}
```

after changing the type area. Maybe you should do this globally using \edef instead of \edef. Most users will never need to change \marginnotetextwidth.

At some environments the vertical adjustment of the margin note will be wrong, e.g. one base line to low. In this case you may use the additional optional argument of `marginnote` at every usage of `marginnote` or redefine \marginnotevadjust at the begin of the environment. The default definition is 0pt.

These macros define how the margin note will be aligned. The defaults are:

- align margin notes at the left margin right to the margin,
- align margin notes at the right margin left to the margin.

You may change this using \renewcommand, e.g. use

```
\renewcommand*{\raggedleftmarginnote}{\centering}
```

to get justified text at the left and centered text at the right margin.

This macro defines the font that will be used to set margin notes. The default is `\normalcolor`. You may use `\renewcommand` to change this, e.g. use

```
\renewcommand*{\marginfont}{\color{red}\sffamily}
```

to get red colored margin notes in sans serif font family. You need to load e.g. package `color` to use `\color`. 
2 Known Problems Using marginnote

At double side layout (e.g. using class option `twoside`) \marginnote needs to know the number of the current page to decide whether the page is odd or even and so whether to use left or right margin. \LaTeX uses an asynchronous output. Because of this counter `page` should not be used to get the number of the current page unless you are at page head or foot. To solve the problem marginnote uses a mechanism similar to labels. But this means, that the correct margin won’t be known at this \LaTeX run but only at the next. So after adding or deleting a margin note or after each change of page break you need two \LaTeX runs to get all margins right.

The command \`marginnote` uses `\strut` and `\vadjust` to put the margin note at the correct position. But under some circumstances this may fail. You may adjust the vertical position of the margin note using the second optional argument of \`marginnote`. Sometimes even the text outside \`marginnote` will be moved because of using \`marginnote`. There’s currently no other solution for this problem then moving the \`marginnote` command.

Note: The margin note will be placed at the current vertical line. This means, if you are using two \`marginnote` commands at the same line, they will be put on the same place. This is not a bug but a feature!

Since release 1.1b \`marginnote` between paragraphs (in vertical mode) will place the note between the paragraphs instead of the end of the previous paragraph. You may use `\leavevmode` or the third optional argument of \`marginnote` to place it different.

No page break may occur inside a margin note created with \`marginnote`. \`marginnote` somewhat different from \`marginpar` if used immediate after `\item`. This is not a bug, it’s a feature!

With math \`marginnote` may work or may not depending on the math environment.

If you are using \TeX, \pdfTeX since version 1.40 or \pdfTeX before version 1.40 with PDF output and the horizontal position of a margin note is wrong, do one more PDF\TeX run.

Sometimes lines are stretched vertically using \`marginnote`, e.g. if you’re using \`marginnote` at a list and upper case umlauts like “Ü” or if \`\lineskiplimit>0pt`. In this case `\lineskiplimit=-\maxdimen` should help.

You should not use \`marginnote` at the optional argument of `\item`.

3 Implementation

First test e-\TeX.

```latex
\begin{group}
\def\@tempb{}\%
\def\@tempa{\%
\PackageError{marginnote}{seems you are not running e-\TeX@\tempb}{%\%
\MessageBreak
```

Since 2004 the LaTeX team recommends to use e-TeX. \MessageBreak
marginnote since version 1.1d uses e-TeX features.

At actual systems ‘latex’ should already use e-TeX.

At deprecated systems it may be called ‘elatex’.

Use either unsupported marginnote up to version 1.1c or ask you administrator for LaTeX using e-TeX.

Not using e-TeX is a fatal error!

Processing cannot be continued!

Next declare and process the options.

Use verbose output mode by default. But you may change this using option quiet.

We need a macro to define a new note at the aux file. This will be done using the mechanism of LATEX that is used for \newlabel. But we use another prefix. This will result in the usual “Labels(s) may have changed. Rerun to get cross-references right.” if a margin note is new or have moved to another page.

We need to know, whether or not PDFTEX and which version of PDFTEX is used. With PDFTEX the horizontal output position may be detected using \pdfsavepos and \pdflastxpos. So the relative position of the margin may be calculated. Without PDFTEX only manual adjustment is available. While PDF mode or not may change before start of the document, setting up the switch is delayed.
Some environments change \textwidth. But at PDF mode we need to know the real text width to find the right margin. So we use our own text width macro. Sometimes it may be useful if the user can set it up. Because of this it is a user command.

\newcommand*{\marginnotetextwidth}{%}
\let\marginnotetextwidth\textwidth
\AtBeginDocument{\if@mn@pdfmode\edef\marginnotetextwidth{\the\textwidth}\fi}

Macro \@mn@margintest does the complete test, which margin to use. The result may be found at \if@tempswa. To avoid changes on the last page if there is a new note on the first page, try to count the notes by page. We know that this can not be successful, but never the less it may be a good try. \@mn@thispage saves the page number of the last usage of \@mn@margintest. \@mn@atthispage

\marginnotetextwidth
saves the number of margin note at this page. But we need to know the absolut page number to do this. So we increase the absolut page number \mn@abspage at every \outputpage. \mn@currpage is the page from the page label if found. \mn@currxpos is somehow special. Using PDF\TeX{} the real $x$ position may be written with the page label and used to calculate the correct horizontal offset. In this case \marginenoteleftadjust and \marginnoterightadjust will not be used!

\begin{verbatim}
78 \newcommand*{\@mn@thispage}{}
79 \newcommand*{\@mn@currpage}{}
80 \newcommand*{\@mn@currxpos}{}
81 \newcounter{mn@abspage}
82 \AtBeginDocument{\setcounter{mn@abspage}{1}%%
83 \g@addto@macro\@outputpage{\stepcounter{mn@abspage}}}
84 \newcommand*{\@mn@margintest}{% 
85 Number of the next margin note at this page.
86 \expandafter\ifx\csname \@mn@thispage\endcsname\@empty
87 \gdef\@mn@thispage{1}%%
88 \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%%
89 \begingroup
90 \@tempcnta\@mn@thispage\advance\@tempcnta by \one
91 \xdef\@mn@atthispage{\the\@tempcnta}%%
92 \endgroup
93 \else
94 \gdef\@mn@atthispage{1}%%
95 \fi
96 \xdef\@mn@thispage{\themn@abspage}%%
\end{verbatim}

Use the number of the page and the number of the margin note at this page to save the real number of this page at the aux file. At PDF mode save the current $x$ position too.

\begin{verbatim}
97 \let\@mn@currpage\relax
98 \let\@mn@currxpos\relax
99 \if\@mn@pdfmode
100 \pdfsavepos
101 \protected@write\@auxout{\let\themn@abspage\relax}{% 
102 \string\newmarginnote{\let\themn@abspage\relax}{% 
103 {\themn@abspage}{\noexpand\number\pdflastxpos sp}}}%
104 \}%
105 \else
106 \protected@write\@auxout{\let\themn@abspage\relax}{% 
107 \string\newmarginnote{\let\themn@abspage\relax}{% 
108 {\themn@abspage}{}}% 
109 \}%
110 \fi
\end{verbatim}

If the margin note label was not defined, it seems to be new. In this case the absolut page number will be used for the test instead of the saved real page number.
Ulrike Fischer suggested a simple change to take care of \hoffset, e.g., using package crop. We use this occasion to take care of \pdfhorigin, too.

\edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\hoffset \relax}%
Command \marginnote is the main macro of the package. The others are helpers to manage the optional arguments.

\newcommand*{\marginnote}{% 
  \@dblarg\@mn@marginnote \newcommand{\@mn@marginnote}{[#2]}{% 
    \ifhmode
      \@bsphack
      \begingroup
      \ifdim\@savsk>\z@\else
        \ifdim\@savsk<\z@\else
          \def\:{\@xifnch}\expandafter\def\: \futurelet\@let@token\@ifnch
        \fi
      \fi
      \else
      \@ifnextchar{\@mn@@marginnote}{[#1]}{#2}}\[
      \fi
    \else
    \begingroup
    \fi
    \@ifnextchar{\@mn@@marginnote}{[#1]}{#2} \[
  \fi
}\newcommand{\@mn@@marginnote}{% 
    \endgroup
  \newcommand{\@mn@@marginnote}{[#1]}{#2}{#3}\[
  \fi
}\endgroup

In horizontal mode the space hack of the \LaTeX kernel will be used. In vertical mode this should not be used.

\ifhmode
  \@mn@vadjust\if@inlabel\leavevmode\fi\ifhmode\strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
\fi
\@tempa{\newcommand{\@mn@marginnote}{()\[}%
  \newcommand{\@mn@marginnote}{#1}{#2}{#3}\[
  \fi
}\else
  \@mn@vadjust\if@inlabel\leavevmode\fi\ifhmode\strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
\fi
\@tempa{\newcommand{\@mn@marginnote}{()\[}%
  \newcommand{\@mn@marginnote}{#1}{#2}{#3}\[
  \fi
}\endgroup

All changes (but change of counters that are global because of using the \LaTeX commands to change them an \gdef and \xdef) should be local. In h-mode a \strut will be used to fix base line. The margin note will be put to vertical list using \vadjust. This also means that we are one line to deep. This will be corrected later using negative kern. In v-mode we use a special kind of vbox to simply set everything. Math mode should behave like v-mode. And if we are just after an item we have to leave v-mode first.

\begingroup
  \ifmmode\strut\let\@tempa\mn@vadjust\else\if@inlabel\leavevmode\fi
  \ifhmode\strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi\fi
  \@tempa{\newcommand{\@mn@marginnote}{#1}{#2}{#3}\[}%
  \newcommand{\@mn@marginnote}{#1}{#2}{#3}\[
  \fi
}\leavevmode
\fi
\@tempa{\newcommand{\@mn@marginnote}{#1}{#2}{#3}\[}%
  \newcommand{\@mn@marginnote}{#1}{#2}{#3}\[
  \fi
}
Everything will be put upwards using a vbox with zero height and depth and \vss.
At this box the margin test will be done. If csreversemargin was used, the logic
switchs. Then the note will be placed to the margin.

\vbox to\z@{\%
\vss
\@mn@margintest
@if@reversemargin@if@tempswa
\@tempswafalse
@else
\@tempswarue
@fi@fi
@if@tempswa
\rlap{\%
If \@mn@currpos is neither \relax nor empty it is the real current $x$ position of
the last PDF\TeX run and may be used to calculate the real horizontal offset.
\iftx\@mn@currxpos\relax
\@kern\@mn@marginnoterightadjust
@if@mn@verbose
\PackageInfo{marginnote}{\%
xpos not known,\MessageBreak
using \string\@mn@marginnoterightadjust}\%
\fi
\else\iftx\@mn@currxpos\@empty
\@kern\@mn@marginnoterightadjust
@if@mn@verbose
\PackageInfo{marginnote}{\%
xpos not known,\MessageBreak
using \string\@mn@marginnoterightadjust}\%
\fi
\else
\if@mn@verbose
\PackageInfo{marginnote}{\%
xpos seams to be \@mn@currxpos,\MessageBreak
\string\@mn@marginnoterightadjust
\space ignored}\%
\fi
\begingroup
\setlength{\@tempdima}{\@mn@currxpos}\%
\@kern\@tempdima
@if@twoside@ifodd\@mn@currpage\relax
\@kern\@oddsidemargin
@else
\@kern\@evensidemargin
@fi
\else
\@kern\@oddsidemargin
\fi
\endgroup
Here's the correction of the vertical position. The rest is simple.

\kern-\parskip
\marginfont\raggedright\hspace{\z@}\strut#2\endgraf
\vss} %
\vss} %
}
}\else
\llap{%
\vbox to\z@{\kern #3
\vbox to\z@{\hsize\marginparwidth
\hspace{\z@}\strut#1\endgraf
\vss} %
\vss} %
\if\@mn@currxpos\relax
\kern-\parskip
\marginfont\raggedleft\hspace{\z@}\strut#1\endgraf
\vss} %
\vss} %
\fi
\if\@mn@currxpos\@empty
\kern\marginnoteleftadjust
\if\@mn@verbose
\PackageInfo{marginnote}{% xpos not known,\MessageBreak using \string\marginnoteleftadjust} %
\fi
\else\if\@mn@currxpos\@empty
\kern\marginnoteleftadjust
\if\@mn@verbose
\PackageInfo{marginnote}{% xpos seams to be \@mn@currxpos,\MessageBreak \string\marginnoteleftadjust space ignored)%
\fi
\else
\if\@mn@verbose
\PackageInfo{marginnote}{% xpos seams to be \@mn@currxpos,\MessageBreak \string\marginnoteleftadjust space ignored)%
\fi
\begingroup
\kern\@mn@currxpos
\if\@twoside\ifodd\@mn@currpage\relax
\kern-\oddsidemargin
\else
\kern-\evensidemargin
\fi
These may be used to define an automatic horizontal adjust. The default is zero. They will be used only if not PDFTeX or PDFTeX before version 1.40 in DVI mode is used, because in this case the save position features are not available.

\newcommand*{\marginnoterightadjust}{\z@}
\newcommand*{\marginnoteleftadjust}{\z@}

This may be used to define an automatic vertical adjust. The default is zero. Values greater than zero will move the margin note down, values less than zero will move the margin note up.

\newcommand*{\marginnotevadjust}{\z@}

This macro is used to set a vertical box without size at vertical mode.

\newcommand{\mn@vlap}[1]{% setbox@tempboxa:vbox to \ht\strutbox#1\vss \box@tempboxa\vskip\baselineskip}

This macro is used to set a horizontal box without height, depth and width.

\newcommand{\mn@zbox}[1]{% bgroup \setbox@tempboxa:vbox\ht\ht\strutbox \dp\dp\strutbox \wd\wd\strutbox \box@tempboxa\egroup}

This macro is used to set a vertical box at horizontal mode.

\newcommand{\mn@vadj}[1]{%
These are very simple. A class may also define \marginfont. Use this if available.
I don’t use \let for the definitions of the ragged macros, so the meaning may
change loading e.g. package ragged2e.

\providecommand*{\marginfont}{}
\newcommand*{\raggedleftmarginnote}{\raggedleft}
\newcommand*{\raggedrightmarginnote}{\raggedright}

Change History

v1.0a
General: Example to macros
\raggedleftmarginnote and
\raggedrightmarginnote at
documentation fixed [thanks to
Susumu Tanimura] ............ 2
\marginfont: Use \providecommand
to define it .................. 12

v1.0b
General: spelling fixes .......... 1

v1.1
\@mn@@@marginnote: new PDF
mode feature .................. 8
\@mn@currpage: new (internal) ... 5
\@mn@currxpos: new (internal) ... 5
\@mn@margintest: new PDF mode
feature .................... 5
\if@mn@pdfmode: new switch .... 4
\marginnotetextwidth: new macro ............... 5

v1.1a
\if@mn@pdfmode: PDF\TeX{} since
1.40 allows \pdfsavepos in DVI
mode too .................... 4

v1.1b
\@mn@@@marginnote: use \mn@vadjust
instead of \vadjust ........... 8
\if@mn@pdfmode: if level fixed .... 4
\mn@vadjust: new (internal) ...... 11
\mn@zbox: new (internal) ...... 11

v1.1c
\if@mn@pdfmode: X\TeX{} has work-
ing \pdflastxpos but no
\pdftexversion ............. 4
\if@mn@pdfmode: X\TeX{} has work-
ing \pdflastxpos but no
\pdftexversion ............. 4
\mn@abspace: take care of \hoffset 7
take care of \pdfhorigin ..... 7