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

Markus Kohm\textsuperscript{‡}

2018/08/09 v1.4b

Abstract

In \LaTeX{} the command \texttt{\marginpar[⟨left⟩]{⟨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{\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{\marginpar}. But there might be other problems . . .

Contents

1 How to Use \texttt{marginnote} Package \hfill 1
2 Known Issues Using \texttt{marginnote} \hfill 3
3 Implementation \hfill 4

1 How to Use \texttt{marginnote} Package

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

\begin{verbatim}
\usepackage{marginnote}
\end{verbatim}

to do so. You may also use one of the following options for a global change of the behaviour of \texttt{marginnote}:

\texttt{fulladjust} adjusts the margin note at the height and depth of the current line.

Note, that this may sometimes result in extra height and depth of the current line, but results in the best vertical alignment. This is the default.

\textsuperscript{*}URL: \url{https://komascript.de/marginnote}

\textsuperscript{†}This file has revision number 27, last revised 2018/08/09. Please note: Currently this package is unmaintained. A new maintainer would be welcome!

\textsuperscript{‡}Email: komascript@gmx.info
heightadjust adjusts the margin note at the height of the current line but not the depth. Note, that this may sometimes result in extra height of the current line and in vertical misplacement.

depthadjust adjusts the margin note at the depth of the current line but not height. Note, that this may sometimes result in extra depth of the current line and very often in vertical misplacement.

noadjust does not adjust the margin note at the height or depth of the current line. Note, that this often results in vertical misplacement but seldom in vertical extra space before or after the current line.

parboxrestore uses a reduced \parboxrestore to restore the definition of \par and \ and several other commands and sets \parindent to 0, \parfillskip to 0pt plus 1fil and \lineskip to \normallineskip, \baselineskip to \normalbaselineskip and activates \sloppy for every margin note. This is the default since release 1.4b of marginnote. Release 1.4 and 1.4a also set \parskip to 0, but this resulted in a vertical placement issue, e.g., inside lists like itemize.

noparboxrestore does not restore any definition for the margin notes. This was the behaviour before release 1.4.

The command \marginnote[(left)]{(right)}{⟨voffset⟩} may be used to set a margin note using marginnote. The first optional argument and the mandatory argument are same using \marginpar from the \LaTeX kernel. Even \reversemarginpar will be considered. The note ⟨left⟩ or ⟨right⟩ will be put at the current vertical position. Second optional argument ⟨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.

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 \xdef instead of \edef. Most users will never need to change \marginnotetextwidth.

\marginnotevadjust 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.

\raggedleftmarginnote 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}{}
\renewcommand*{\raggedrightmarginnote}{\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.

\section{Known Issues Using marginnote}

Currently the package lacks of a new maintainer. The author does not longer
maintain the package, because originally it was designed to be only a small hack to
be used on some cases. But over the years it has grown and became a conglomerate
of hacks that hack hacks that are used very widely. A break with a complete revise
of the design would be necessary. But nothing I can or want to do currently. If
you want to do it, please contact the author (see the footnotes on the first page).

You can not use paragraphs with \par or empty lines inside a margin note. In
my opinion this would note make sense. However, if you really need paragraphs
you can try to use \endgraf instead of \par.

From version 1.4a there is a workaround for double-ended documents with
consecutive odd pages or consecutive even pages. However it is not recommended
to use double-ended documents with such page sequences, because printing such
documents could be a mess. marginnote shows a warning message whenever it
detects those page sequences.

From version 1.3 marginnote does not longer support \TeX engines without
primitives \pdfsavepos/\savepos and \pdflastxpos/\lastxpos. The former
(manual adjustment) fallback has been removed. You’ll get an error message, if
you try to use a \TeX engine without these primitives. Also \vTeX primitive are
needed. However, with current free \TeX distributions like MiK\TeX or \TeXLive
this shouldn’t be a problem.

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. You may use one of the package options fulladjust, heightadjust, depthadjust or noadjust to change the global adjustment or a local redefinition of \mn@strut or \mn@zbox.

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 is 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 Xe\TeX, PDF\LaTeX since version 1.40 or PDF\LaTeX before version 1.40 with PDF output and the horizontal position of a margin note is wrong, do one more PDF\LaTeX run.

Sometimes lines are stretched vertically using \marginnote, e.g. if you’re using \marginnote at a list and upper case umlauts like “¨U” or if \lineskiplimit>0pt. In this case \lineskiplimit=0pt or \lineskiplimit=-\maxdimen, or one of the options may help.

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

If \if@twocolumn is \iftrue, e.g., because you are using option twocolumn or command \twocolumn, \marginnote does decide whether the note should be placed left of the column or right of the columns simply by comparing the current horizontal position with \columnwidth+\columnsep. So if the current horizontal position is somewhere in the left column, the note is placed in the left margin. If the current horizontal position is somewhere right of the left column, the note is placed in the right margin. However, support for twocolumn mode is as problematic as support for reverse margin notes. I do not like it. Maybe it will be changed in future. The current support for twocolumn mode has been implemented only because of a feature request by Florent Chervet.

3 Implementation

Since version 1.3 \marginnote does need either \pdfsavepos and \pdflastxpos or \savepos and \lastxpos and does not longer support engines without these primitives. All these engines also provide \v-\TeX extensions. So we do not longer need an explicite \v-\TeX test.

1 \begingroup
2 \@ifundefined{pdfsavepos}{%
3 \@ifundefined{savepos}{%
Next declare and process the options.

\if@mn@verbose
\newif\if@mn@verbose\@mn@verbostrue
\DeclareOption{verbose}{\@mn@verbostrue}
\DeclareOption{quiet}{\@mn@verbostruefalse}
\fi

\mn@strut
The package needs to adjust the margin note at the current line. Sometimes this
causes extra vertical line spacing. To avoid this you may redefine \mn@strut. The
default value is \strut.
\newcommand{\mn@strut}{}

\mn@zbox
This macro is used to set a horizontal box without height, depth and width.
\newcommand{\mn@zbox}[1]{}
The options do redefine both, \mn@strut and \mn@zbox.

\DeclareOption{fulladjust}{% 
\renewcommand{\mn@strut}{\strut}
\renewcommand{\mn@zbox}[1]{% 
\bgroup 
\setbox\@tempboxa\vbox{#1} 
\ht\@tempboxa\ht\strutbox 
\dp\@tempboxa\dp\strutbox 
\wd\@tempboxa\z@ 
\box\@tempboxa 
\egroup 
}} %
\DeclareOption{heightadjust}{% 
\renewcommand{\mn@strut}{\begingroup\dp\strutbox\z@\strut\endgroup} 
\renewcommand{\mn@zbox}[1]{% 
\bgroup 
\setbox\@tempboxa\vbox{#1} 
\ht\@tempboxa\ht\z@ 
\dp\@tempboxa\dp\strutbox 
\wd\@tempboxa\z@ 
\box\@tempboxa 
\egroup 
}} %
\DeclareOption{depthadjust}{% 
\renewcommand{\mn@strut}{\begingroup\ht\strutbox\z@\strut\endgroup} 
\renewcommand{\mn@zbox}[1]{% 
\bgroup 
\setbox\@tempboxa\vbox{#1} 
\ht\@tempboxa\ht\z@ 
\dp\@tempboxa\dp\z@ 
\wd\@tempboxa\z@ 
\box\@tempboxa 
\egroup 
}} %
\DeclareOption{fulladjust}{% 
\renewcommand{\mn@strut}{\strut}
\renewcommand{\mn@zbox}[1]{% 
\bgroup 
\setbox\@tempboxa\vbox{#1} 
\ht\@tempboxa\ht\strutbox 
\dp\@tempboxa\dp\strutbox 
\wd\@tempboxa\z@ 
\box\@tempboxa 
\egroup 
}} %

\newcommand*{\newmarginnote}{\@newl@bel{mn}}
\if@mn@pdfmode
  \@mn@mode@prefix
\fi
\AtBeginDocument{\edef\marginnotetextwidth{\the\textwidth}}

\begin{Verbatim}
\texttt{\newmarginnote} \texttt{\textwidth}
\end{Verbatim}

\begin{Verbatim}
\texttt{\newcommand*{\marginnotetextwidth}{}}
\end{Verbatim}

\begin{Verbatim}
\texttt{\let\marginnotetextwidth=\textwidth}
\end{Verbatim}

\begin{Verbatim}
\texttt{\AtBeginDocument{\edef\marginnotetextwidth{\the\textwidth}}}
\end{Verbatim}

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 \texttt{\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.

\begin{Verbatim}
\texttt{\newcommand*{\newmarginnote}{\@newl@bel{mn}}}\}
\end{Verbatim}

\begin{Verbatim}
\texttt{\if@mn@pdfmode}
\end{Verbatim}

\begin{Verbatim}
\texttt{\texttt{\@mn@mode@prefix}}
\end{Verbatim}

\begin{Verbatim}
\texttt{\texttt{\marginnotetextwidth}}\}
\end{Verbatim}

Some environments change \texttt{\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.

\begin{Verbatim}
\texttt{\newcommand*{\marginnotetextwidth}{}}\}
\end{Verbatim}

\begin{Verbatim}
\texttt{\let\marginnotetextwidth=\textwidth}
\end{Verbatim}

\begin{Verbatim}
\texttt{\AtBeginDocument{\edef\marginnotetextwidth{\the\textwidth}}}
\end{Verbatim}
Macro \texttt{\@m@marginintest} does the complete test, which margin to use. The result may be found at \texttt{\@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 cannot be successful, but never the less it may be a good try. \texttt{\@m@thispage} saves the page number of the last usage of \texttt{\@m@marginintest}. \texttt{\@m@atthispage} saves the number of margin note at this page. But we need to know the absolute page number to do this. So we increase the absolute page number \texttt{\@m@abspage} at every \texttt{\@outputpage}. \texttt{\@m@currpage} is the page from the page label if found. \texttt{\@m@currxpos} is the real $x$ position may be written with the page label and used to calculate the correct horizontal offset.

\newcommand*{\@m@thispage}{}
\newcommand*{\@m@currpage}{}
\newcommand*{\@m@currxpos}{}
\newcounter{\m@abspage}
\AtBeginDocument{\setcounter{\m@abspage}{1}%
  \g@addto@macro\@outputpage{\
    \stepcounter{\m@abspage}\
  }%
  \PackageWarningNoLine{marginnote}{Consecutive odd pages found.\
    Note, it is not recommended to use consecutive\
    odd pages in a double-ended document.\
    The pages of your document should always\
    be a sequence: odd-even-odd-even-...\
    \PackageWarningNoLine{marginnote}{Maybe you’ve forgotten a\
    \@ifundefined{KOMAClassName}{}{\string\cleardoublepage}\
    \@ifundefined{KOMAClassName}{}{\string\cleardoubleoddpage}\
    before\MessageBreak\
    changing the page numbering on page \thepage}\
}%
\endgroup
\fi
\PackageInfo{marginnote}{Using workaround for absolute page number}%
\stepcounter{\m@abspage}%
\fi
\else
  \if@twoside
    \begingroup
    \PackageWarningNoLine{marginnote}{Consecutive odd pages found.\
    Note, it is not recommended to use consecutive\
    odd pages in a double-ended document.\
    The pages of your document should always\
    be a sequence: odd-even-odd-even-...\
    \PackageWarningNoLine{marginnote}{Maybe you’ve forgotten a\
    \@ifundefined{KOMAClassName}{}{\string\cleardoublepage}\
    \@ifundefined{KOMAClassName}{}{\string\cleardoubleoddpage}\
    before\MessageBreak\
    changing the page numbering on page \thepage}\
  )%\endgroup
  \fi
\else
  \ifodd\value{\m@abspage}%
    \ifodd\value{page}%
      \else
        \PackageWarningNoLine{marginnote}{Using workaround for absolute page number}%
        \stepcounter{\m@abspage}%
      \fi
    \fi
  \fi
Consecutive even pages found.

Note, it is not recommended to use consecutive even pages in a double-ended document.

The pages of your document should always be a sequence: odd-even-odd-even-...

Maybe you’ve forgotten a \ifundefined{KOMAClassName}{}
\cleardoublepage{}
\cleardoubleevenpage{}

before changing the page numbering on page \thepage

\stepcounter{mn@abspage}
\fi
\fi
\fi
\newcommand*{\@mn@margintest}{
Number of the next margin note at this page.
\expandafter\ifx\csname @mn@thispage\endcsname\@empty
\def\@mn@atthispage{1}
\else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}
\begin{group}
\@tempcnta\@mn@atthispage\advance\@tempcnta by 1
\xdef\@mn@atthispage{\the\@tempcnta}
\end{group}
\else
\def\@mn@atthispage{1}
\fi
\fi
\xdef\@mn@thispage{\themn@abspage}
\xdef\@mn@thispage{\themn@abspage}
\edef\themn@abspage{\relax}
\edef\@mn@lastxpos{\relax}
\protected@write\@auxout{\let\themn@abspage{\relax}{%}
\string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%}
\{\themn@abspage}{\noexpand\number\@mn@lastxpos sp}}%
\expandafter\ifx\csname mn@note.\@mn@thispage.\@mn@atthispage\endcsname\relax
If we are not in two side mode, we are on an odd page.

\if@twoside
  \if@mn@verbose
    \PackageInfo{marginnote}{Suggest that margin note \mn@thispage, \mn@atthispage space will be on \MessageBreak absolute page \themn@abspage. \MessageBreak This may be wrong}\
  \fi
  \ifodd\value{mn@abspage}\@tempswatrue\else\@tempswafalse\fi
\else
  \if@mn@verbose
    \PackageInfo{marginnote}{right page because not two side mode}\
  \fi
  \@tempswatrue
\fi
\else
  \edef\@mn@currpage{\csname mn@note.\mn@thispage.\mn@atthispage\endcsname}\
  \edef\@mn@currxpos{\expandafter\@secondoftwo\@mn@currpage}\
  Ulrike Fischer suggested a simple change to take care of \hoffset, e.g., using package \texttt{crop}. We use this occasion to take care of \texttt{pdfhorigin}, too. If \@mn@currxpath is not empty here, it should be corrected by \hoffset and maybe by \texttt{pdfhorigin}.

\ifeq\@mn@currxpath@empty\else
  \edef\@mn@currxpath{\the\dimexpr\@mn@currxpath - \hoffset\relax}\
  \begingroup\expandafter\expandafter\expandafter\endgroup
    \ifx\csname pdfhorigin\endcsname\relax
      \begingroup\expandafter\expandafter\expandafter\endgroup
        \ifx\csname pdfoutput\endcsname\relax
          \begingroup\expandafter\expandafter\expandafter\endgroup
            \ifnum\outputmode=1 %
              \edef\@mn@currxpath{\the\dimexpr\@mn@currxpath - \pdfhorigin +1in}\relax
            \fi
          \fi
        \fi
      \fi
    \fi
  \fi
\fi
\ifdefined\mn@pagewidth
  \edef\@mn@currxpath{\the\dimexpr\@mn@currxpath - \pdfhorigin +1in}\relax
\fi
\fi

If you are using package \texttt{bidi} and RTL mode is active, the position is from right instead of left. So we have to subtract \@mn@currxpath from \texttt{pdffpagewidth} (or \texttt{pagewidth} using \LaTeX{}, but this cannot be, because \texttt{bidi} is not \LaTeX{}-compatible).

\ifdef\mn@pagewidth
\@mn@if@RTL{%
\PackageInfo{marginnote}{Margin note \@mn@thispage. \@mn@atthispage\space in RTL mode}%
\edef\@mn@currpos{%
 \the\dimexpr\mn@pagewidth-\@mn@currpos\relax
}
\fi
\fi
\edef\@mn@currpage{\expandafter\@firstoftwo\@mn@currpage}%
\if@mn@verbose
\PackageInfo{marginnote}{Margin note \@mn@thispage. \@mn@atthispage\space is on absolute page \@mn@currpage}%
\fi
\if@twoside
\ifodd\@mn@currpage\relax
\@tempswatrue
\if@twocolumn
\ifdim\@mn@currpos<\dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
\@tempswafalse
\fi
\fi
\else
\@tempswafalse
\if@twocolumn
\ifdim\@mn@currpos>\dimexpr\evensidemargin+\columnwidth\relax
\@tempswatrue
\fi
\fi
\else
\@tempswafalse
\if@twocolumn
\ifdim\@mn@currpos<\dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
\@tempswafalse
\fi
\fi
\else
\@tempswafalse
\if@mn@verbose
\PackageInfo{marginnote}{right page because not two side mode}%
\fi
\@tempswatrue
\if@twocolumn
\ifdim\@mn@currpos<\dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
\@tempswafalse
\fi
\fi
\else
\@tempswafalse
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\if@mn@verbose
\PackageInfo{marginnote}{right page because not two side mode}%
\fi
\@tempswatrue
\if@twocolumn
\ifdim\@mn@currpos<\dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
\@tempswafalse
\fi
\fi
\else
\@tempswafalse
\if@mn@verbose
\PackageInfo{marginnote}{right page because not two side mode}%
\fi
\@tempswatrue
\if@twocolumn
\ifdim\@mn@currpos<\dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
\@tempswafalse
\fi
\fi
\else
\@tempswafalse
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\@mn@ifRTL Test, whether or not \if@RTL exists and is true or false.
\marginnote 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\def:{\@xifnch}\expandafter\def:{\futurelet\@let@token\@ifnch}% \fi\else\begingroup\fi\@ifnextchar[{\@mn@@marginnote[{#1}]{#2}}{\@mn@@marginnote[{#1}]{#2}\[\z@]}% \newcommand{\@mn@@marginnote}{}\long\def{\@mn@@marginnote}[#1]{\endgroup\fi\@ifnextchar[{{\@mn@@@marginnote[{#1}]{#2}}{\@mn@@@marginnote[{#1}]{#2}\[\z@]}%}
\long\def{\@mn@@@marginnote}[#1]{% 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\mn@strut\let\@tempa\mn@vadjust\else\if@inlabel\leavevmode\fi\ifhmode\mn@strut\let\@tempa\mn@vlap\fi\@ifnextchar[{{\@mn@@@marginnote[{#1}]{#2}}{\@mn@@@marginnote[{#1}]{#2}\[\z@]}%}}{\@mn@@@marginnote[{#1}]{#2}\[\z@]}%}
\long\def{\@mn@@@marginnote}[#1]{%}}{\@mn@@@marginnote}[#1]{%}}\long\def{\@mn@@@marginnote}[#1]{%}}\begin{document}
\begin{itemize}
\item This is a list item.
\item Another list item.
\end{itemize}
\end{document}
\fi
\@tempa{%
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 \reversemarginpar was used, the logic reverses. Then the note will be places to the margin.
\vbox to\z@{%\vss
\@m@nm@argin@test
\if@reversemargin\if@tempswa
\@tempswafalse
\else
\@tempswatrue
\fi\fi
\if@tempswa
\rlap{If \@m@ncurr@x@p@os is neither \relax nor empty it is the real current x position of the last PDF\LaTeX run and may be used to calculate the real horizontal offset.
\if@m@n@ver@bose
\PackageInfo{marginnote}{xpos seems to be \@m@ncurr@x@p@os}%
\fi
\begingroup
\ifx\@m@ncurr@x@p@os\relax\else\ifx\@m@ncurr@x@p@os\@empty\else
\kern-\dimexpr\@m@ncurr@x@p@os\relax
\fi\fi
\if@t@w@side\ifodd\@m@ncurr@pa@ge\relax
\kern\oddsidemargin
\else
\kern\evensidemargin
\fi
\kern 1in
\endgroup
\kern\marginnotetextwidth\kern\marginparsep
\vbox to\z@{\kern\marginnotevadjust\kern #3
\vbox to\z@{%\hsize\marginparwidth
\linewidth\hsize
Here’s the correction of the vertical position. The remain is simple.
\kern-\parskip
\m@parboxre@store
\m@nfont\raggedrightmarginnote\strut\hspace{\z@}%
\ignorespaces#2\endgraf
}
Using the left margin.

\llap{\vbox to\z@{\kern\marginnotevadjust\kern #3\vbox to\z@{\hsize\marginparwidth}\linewidth\hsize\hsize\linewidth\hspace{\z@}\ignorespaces#1\endgraf\vss\vss}}\if@mn@verbose\PackageInfo{marginnote}{xpos seems to be \@mn@currxpos}\fi\begingroup\ifx\@mn@currxpos\relax\else\ifx\@mn@currpos\@empty\else\kern\@mn@currxpos\fi\fi\if\@mn@currpage\relax\let\@mn@currpage\@ne\fi\if@twoside\ifodd\@mn@currpage\relax\kern-\oddsidemargin\else\kern-\evensidemargin\fi\kern-1in\endgroup\kern\marginparsep\marginnoterightadjust\marginnoteleftadjust
\margintevadjust 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.

\mn@vlap This macro is used to set a vertical box without size at vertical mode.

\mn@vadjust This macro is used to set a vertical box at horizontal mode.

\marginfont 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.

\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.

\mfont \mnote \mnoteleft \mnoteright \textwidth \vadjust

\changehistory v1.0a \providecommand to define it. 15
\mfont: Use General: Example to macros
\raggedleftmarginnote \and \raggedrightmarginnote at documentation fixed [thanks to Susumu Tanimura].

v1.0b
General: spelling fixes

v1.1
\@mn@marginnote: new PDF mode feature \ignorespaces added 
\@mn@currxpos: new (internal) \ignorespaces added 
\@mn@currpage: new (internal) \ignorespaces added 
\@mn@margintest: new PDF mode feature \ignorespaces added
\if@mn@pdfmode: new switch \ignorespaces added 
\marginnotetextwidth: new \ignorespaces added

v1.1a
\if@mn@pdfmode: PDF\TeX \ignorespaces added 
1.40 allows \pdflastxpos in DVI mode too \ignorespaces added

v1.1b
\@mn@marginnote: use \ignorespaces added 
\@mn@vadjust instead of \vadjust \ignorespaces added 
\if@mn@pdfmode: if level fixed \ignorespaces added 
\@mn@vadjust: new switch \ignorespaces added 
\@mn@zbox: new \ignorespaces added

v1.1c
\if@mn@pdfmode: \ignorespaces added 
X\TeX has working \pdflastxpos \ignorespaces added

v1.1d
\@mn@abspage: take care of \ignorespaces added 
hoffset \ignorespaces added 
take care of \pdflastxpos \ignorespaces added

v1.1e
\@mn@marginnote: use \ignorespaces added 
\@mn@strut instead of \strut \ignorespaces added 
\@mn@strut: new (semi internal) \ignorespaces added 
General: new options fulladjust, heighthadjust, depthadjust, and noadjust \ignorespaces added

v1.1f
\@mn@marginnote: missing usage of \marginnotevadjust on left margin fixed \ignorespaces added

v1.1g
\@mn@marginnote: missing \ignorespaces added 
\long added \ignorespaces added 
set \linewidth \ignorespaces added 
\@mn@marginnote: missing \long \ignorespaces added 

v1.1i
\@mn@marginnote: \ignorespaces added 
\ignorespaces added 
\strut moved to fix hyphenation (thanks to Ulrike Fischer) \ignorespaces added

v1.2
\@mn@mode@prefix: (new (internal) \ignorespaces added
\if@mn@pdfmode: addition for \lua\TeX \ignorespaces added 
from 0.85 \ignorespaces added 
\@mn@abspage: addition for \lua\TeX \ignorespaces added 
from 0.85 \ignorespaces added

v1.2a
\@mn@abspage: redefine \ignorespaces added 
\@mn@abspage only if not empty \ignorespaces added

v1.2b
\@mn@ifRTL: new internal \ignorespaces added 
\@mn@abspage: bidi code added \ignorespaces added 
General: spelling fixes (by Thomas Reuben) \ignorespaces added

v1.3
\@mn@mode@prefix: removed \ignorespaces added 
\if@mn@pdfmode: removed \ignorespaces added 
\@mn@abspage: removed \ignorespaces added 
\@mn@marginnoteleftadjust: removed \ignorespaces added 
\@mn@marginnoterightadjust: removed \ignorespaces added 
\@mn@abspage: twocolumn \ignorespaces added 
test \ignorespaces added 
added \ignorespaces added 
non PDF mode removed \ignorespaces added 
use new internals \@mn@savepos \ignorespaces added 
and \@mn@lastxpos \ignorespaces added \@mn@lastxpos \ignorespaces added 
new internal \ignorespaces added 
command \ignorespaces added 
\@mn@savepos: new internal \ignorespaces added 
command \ignorespaces added 
General: \epsilon\TeX \ignorespaces added 
removed \ignorespaces added 
early \pdflastxpos/\savepos \ignorespaces added 
test \ignorespaces added

v1.4
\@mn@marginnote: \ignorespaces added 
\@mn@parboxrestore \ignorespaces added 
\@mn@parboxrestore: new internal \ignorespaces added 
command \ignorespaces added 
General: new options \parboxrestore (default) \ignorespaces added 
and \noparboxrestore \ignorespaces added 
\@mn@abspage: workaround for screwball page order \ignorespaces added