Non-Floating Margin Notes with marginnote Package*

Markus Kohm†

2010/01/05

Abstract

In LaTeX the command `\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 *marginnote* supports another command `\marginnote` to create notes in the margin. This does not use a kind of float and for this does not have the disadvantage of `\marginpar`. But there might be other problems . . .

Contents

1 How to Use marginnote Package

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

```
\usepackage{marginnote}
```

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

`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.

*This file has version number v1.1f, last revised 2010/01/05.

†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.

`\marginnote`

The command `\marginnote[`⟨*left*⟩`]{`⟨*right*⟩`}[`⟨*voffset*⟩`]` my 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.

`\marginnoteleftadjust`
`\marginnoterightadjust`

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

```
\begingroup
  \makeatletter
  \g@addto@macro\framed{%
    \let\marginnoteleftadjust\FrameSep
    \let\marginnoterightadjust\FrameSep
  }
\endgroup
```

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 PDFTeX features (`\pdfsavepos` and `\pdflastxpos`). So if you are using PDFLaTeX with PDF output or PDFLaTeX with PDFTeX-version since 1.40 or XeTeX you will not need to use the example code above, but you will need at least two LaTeX runs to get the correct horizontal positions of the margin notes.

`\marginnotetextwidth`

Package *marginnote* needs to know the real width of the type area to find the right margin. While some environments (e.g. thos of package *framed*) change `\textwidth`, *marginnote* defines it's 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
\raggedrightmarginnote

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.

\marginfont

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 wether the page is odd or even and so wether 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, depthajust or noajust 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 paragaphs 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 occure 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 XƎTEX, PDFLATEX since version 1.40 or PDFLATEX before version 1.40 with PDF output and the horizontal position of a margin note is wrong, do one more PDFLATEX 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=0pt` or `\lineskiplimit=-\maxdimen`, or one of the options may help.

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

3 Implementation

First test ε-TEX.

```
\begingroup
  \def\@tempb{}%
  \def\@tempa{%
    \PackageError{marginnote}{seems you are not running e-TeX\@tempb}{%
      Since 2004 the LaTeX team recommends to use e-TeX.\MessageBreak
      marginnote since version 1.1d uses e-TeX features.\MessageBreak
      At actual systems `latex' should already use e-TeX.\MessageBreak
      At deprecated systems it may be called `elatex'.\MessageBreak
      Use either unsupported marginnote up to version 1.1c or\MessageBreak
      ask you administrator for LaTeX using e-TeX\@tempb.\MessageBreak
      Not using e-TeX\@tempb\space is a fatal error!\MessageBreak
      Processing cannot be continued!}%
    \endgroup
    \batchmode \errmessage{}\csname @@end\endcsname\end\relax
    \csname endinput\endcsname
  }%
  \expandafter\ifx\csname eTeXversion\endcsname\relax\else
    \ifnum\eTeXversion <2
      \def\@tempb{ V 2}%
    \else
      \let\@tempa\endgroup
    \fi
  \fi
\@tempa
```

Next declare and process the options.

`\if@mn@verbose` Use verbose output mode by default. But you may change this using option `quiet`.

```
\newif\if@mn@verbose\@mn@verbosetrue
```

```
\DeclareOption{verbose}{\@mn@verbosetrue}
\DeclareOption{quiet}{\@mn@verbosefalse}
```

\mn@strut The package needs to adjust the margin note at the current line. Sometimes this provocates 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\strutbox
      \dp\@tempboxa\dp\z@
      \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\strutbox
      \wd\@tempboxa\z@
      \box\@tempboxa
    \egroup
  }%
}
\DeclareOption{noadjust}{%
```

```
  \renewcommand*{\mn@strut}{\relax}%
  \renewcommand{\mn@zbox}[1]{%
    \bgroup
      \setbox\@tempboxa\vbox{\kern-\ht\strutbox #1}%
      \ht\@tempboxa\ht\z@
      \dp\@tempboxa\dp\z@
      \wd\@tempboxa\z@
      \box\@tempboxa
    \egroup
  }%
}
```

```
\ExecuteOptions{verbose,fulladjust}
\ProcessOptions\relax
```

`\newmarginnote` 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.

```
\newcommand*{\newmarginnote}{\@newl@bel{mn}}
```

`\if@mn@pdfmode` We need to know, wether 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.

```
\newif\if@mn@pdfmode\@mn@pdfmodefalse
\AtBeginDocument{%
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname pdflastxpos\endcsname\relax\else % bg or 1
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname pdftexversion\endcsname\relax % bg 2
      \begingroup\expandafter\expandafter\expandafter\endgroup
      \expandafter\ifx\csname pdfoutput\endcsname\relax % bg 3
        \begingroup\expandafter\expandafter\expandafter\endgroup
        \expandafter\ifx\csname XeTeXrevision\endcsname\relax\else % bg 4
          \@mn@pdfmodetrue
        \fi % ed 4
      \else % or 3
        \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi % bg ed 4
      \fi % ed 3
    \else % or 2
      \ifnum \pdftexversion<140 % bg 3
        \begingroup\expandafter\expandafter\expandafter\endgroup
        \expandafter\ifx\csname pdfoutput\endcsname\relax % bg 4
        \else % or 4
          \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi % bg ed 5
        \fi % ed 4
      \else % or 3
```

```
        \@mn@pdfmodetrue
      \fi % ed 3
    \fi % ed 2
  \fi % ed 1
  \if@mn@verbose
    \if@mn@pdfmode
      \PackageInfo{marginnote}{%
        \string\pdfoutput\space not 0 and \string\pdflastxpos\space
        available.\MessageBreak
        Extended position detection mode activated\@gobble
      }%
    \else
      \PackageInfo{marginnote}{%
        either \string\pdflastxpos\space or \string\pdfoutput\space not
        available\MessageBreak
        or \string\pdfoutput\space set to 0.\MessageBreak
        Extended position detection mode deactivated\@gobble
      }%
    \fi
  \fi
}
```

`\marginnotetextwidth`

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 usefull 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}
```

`\@mn@margintest`
`\@mn@thispage`
`\@mn@atthispage`
`\@mn@currpage`
`\@mn@currxpos`
`mn@abspage`

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 successfull, 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` 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 PDFTeX the real x position may be written with the page label and used to calculate the correct horizontal offset. In this case `\marginnoteleftadjust` and `\marginnoterightadjust` will not be used!

```
\newcommand*{\@mn@thispage}{}
\newcommand*{\@mn@currpage}{}
\newcommand*{\@mn@currxpos}{}
\newcounter{mn@abspage}
\AtBeginDocument{\setcounter{mn@abspage}{1}%
  \g@addto@macro\@outputpage{\stepcounter{mn@abspage}}}
\newcommand*{\@mn@margintest}{%
```

Number of the next margin note at this page.

```
\expandafter\ifx\csname @mn@thispage\endcsname\@empty
  \gdef\@mn@atthispage{1}%
\else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
    \begingroup
      \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
      \xdef\@mn@atthispage{\the\@tempcnta}%
    \endgroup
  \else
    \gdef\@mn@atthispage{1}%
  \fi
\fi
\xdef\@mn@thispage{\themn@abspage}%
```

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.

```
\let\@mn@currpage\relax
\let\@mn@currxpos\relax
\if@mn@pdfmode
  \pdfsavepos
  \protected@write\@auxout{\let\themn@abspage\relax}{%
    \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
      {\themn@abspage}{\noexpand\number\pdflastxpos sp}}%
  }%
\else
  \protected@write\@auxout{\let\themn@abspage\relax}{%
    \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
      {\themn@abspage}{}}%
  }%
\fi
```

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.

```
\expandafter\ifx\csname mn@note.\@mn@thispage.\@mn@atthispage\endcsname\relax
```

If we are not in two side mode, we are on a 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 crop. We use this occasion to take care of \pdfhorigin, too.

```
    \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\hoffset\relax}%
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname pdfhorigin\endcsname\relax\else
      \begingroup\expandafter\expandafter\expandafter\endgroup
      \expandafter\ifx\csname pdfoutput\endcsname\relax\else
        \ifnum \pdfoutput=1 %
          \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
            +1in\relax}%
        \fi
      \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\MessageBreak}%
    \fi
    \if@twoside
      \ifodd\@mn@currpage\relax
        \@tempswatrue
      \else
        \@tempswafalse
      \fi
    \else
      \if@mn@verbose
        \PackageInfo{marginnote}{right page because not two side mode}%
      \fi
      \@tempswatrue
    \fi
  \fi
}
```

\marginnote
\@mn@marginnote
\@mn@@marginnote
\@mn@@@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}{}
\def\@mn@@marginnote[#1]#2[#3]{%
  \endgroup
```

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

```
  \ifhmode
    \@mn@@@marginnote[{#1}]{#2}[{#3}]%
    \@esphack
  \else
    \@mn@@@marginnote[{#1}]{#2}[{#3}]%
  \fi
}
\newcommand{\@mn@@@marginnote}{}
\def\@mn@@@marginnote[#1]#2[#3]{%
```

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 wie are one line to deep. This will be corrected later using negative kern. In v-mode wie 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@vadjust\else\let\@tempa\mn@vlap\fi
    \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 csreversemargin was used, the logic switchs. Then the note will be places to the margin.

```
      \vbox to\z@{%
        \vss
        \@mn@margintest
        \if@reversemargin\if@tempswa
            \@tempswafalse
          \else
            \@tempswatrue
        \fi\fi
        \if@tempswa
          \rlap{%
```

If `\@mn@currpos` is neither `\relax` nor empty it is the real current x position of the last PDFLaTeX run and may be used to calculate the real horizontal offset.

```
            \ifx\@mn@currxpos\relax
              \kern\marginnoterightadjust
              \if@mn@verbose
                \PackageInfo{marginnote}{%
                  xpos not known,\MessageBreak
                  using \string\marginnoterightadjust}%
              \fi
            \else\ifx\@mn@currxpos\@empty
                \kern\marginnoterightadjust
                \if@mn@verbose
                  \PackageInfo{marginnote}{%
                    xpos not known,\MessageBreak
                    using \string\marginnoterightadjust}%
                \fi
              \else
                \if@mn@verbose
                  \PackageInfo{marginnote}{%
                    xpos seems to be \@mn@currxpos,\MessageBreak
                    \string\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
                  \kern 1in
                \endgroup
              \fi
            \fi
            \kern\marginnotetextwidth\kern\marginparsep
            \vbox to\z@{\kern\marginnotevadjust\kern #3
              \vbox to\z@{%
                \hsize\marginparwidth
```

Here's the correction of the vertical position. The rest is simple.

```
                \kern-\parskip
                \marginfont\raggedrightmarginnote\hspace{\z@}\strut#2\endgraf
                \vss}%
              \vss}%
          }%
        \else
```

Using the left margin.

```
     \llap{%
      \vbox to\z@{\kern\marginnotevadjust\kern #3
       \vbox to\z@{%
        \hsize\marginparwidth
```

Same like above for left margins.

```
        \kern-\parskip
        \marginfont\raggedleftmarginnote\hspace{\z@}\strut#1\endgraf
        \vss}%
       \vss}%
      \ifx\@mn@currxpos\relax
       \kern\marginnoteleftadjust
       \if@mn@verbose
        \PackageInfo{marginnote}{%
         xpos not known,\MessageBreak
         using \string\marginnoteleftadjust}%
       \fi
      \else\ifx\@mn@currxpos\@empty
        \kern\marginnoteleftadjust
        \if@mn@verbose
         \PackageInfo{marginnote}{%
          xpos not known,\MessageBreak
          using \string\marginnoteleftadjust}%
        \fi
       \else
        \if@mn@verbose
         \PackageInfo{marginnote}{%
          xpos seems 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
         \else
          \kern-\oddsidemargin
         \fi
         \kern-1in
        \endgroup
       \fi
      \fi
      \kern\marginparsep
     }%
    \fi
   }%
```

```
    }%
  \endgroup
}
```

\marginnoterightadjust
\marginnoteleftadjust

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

\marginnotevadjust

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

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

\mn@vlap

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
}
```

\mn@vadjust

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

```
\newcommand{\mn@vadjust}[1]{%
  \mn@zbox{\kern-\parskip
    \leavevmode\vadjust{#1}%
    \kern\parskip
  }%
}
```

\marginfont
\raggedleftmarginnote
\raggedrightmarginnote

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]. 3

`\marginfont`: Use `\providecommand` to define it. 13

v1.0b

General: spelling fixes 1

v1.1
- `\@mn@@@marginnote`: new PDF mode feature 9
- `\@mn@currpage`: new (internal) . . . 7
- `\@mn@currxpos`: new (internal) . . . 7
- `\@mn@margintest`: new PDF mode feature 7
- `\if@mn@pdfmode`: new switch 6
- `\marginnotetextwidth`: new macro 7

v1.1a
- `\if@mn@pdfmode`: PDFTEX since 1.40 allows `\pdfsavepos` in DVI mode too 6

v1.1b
- `\@mn@@@marginnote`: use `\mn@vadjust` instead of `\vadjust` 10
- `\if@mn@pdfmode`: if level fixed 6
- `\mn@vadjust`: new (internal) 13
- `\mn@zbox`: new (internal) 5

v1.1c
- `\if@mn@pdfmode`: XƎTEXhas working `\pdflastxpos` but no `\pdftexversion` 6

v1.1d
- `mn@abspage`: take care of `\hoffset` 9
  - take care of `\pdfhorigin` 9

v1.1e
- `\@mn@@@marginnote`: use `\mn@strut` instead of `\strut` 10
- General: new options `fulladjust`, `heightadjust`, `depthadjust`, and `noadjust` 5
- `\mn@strut`: new (semi internal) . . 5

v1.1f
- `\@mn@@@marginnote`: missing usage of `\marginnotevadjust` on left margin fixed 12