

# Non-Floating Margin Notes with `marginnote` Package\*

Markus Kohm<sup>†</sup>

2018/04/13 v1.3

## Abstract

In L<sup>A</sup>T<sub>E</sub>X 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

<b>1</b>	<b>How to Use <code>marginnote</code> Package</b>	<b>1</b>
<b>2</b>	<b>Known Issues Using <code>marginnote</code></b>	<b>3</b>
<b>3</b>	<b>Implementation</b>	<b>4</b>

## 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 revision number 21, last revised 2018/04/20.

<sup>†</sup>Email: [komascript@gmx.info](mailto: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⟩]` may be used to set a margin note using `\marginnote`. The first optional argument and the mandatory argument are same using `\marginpar` from the L<sup>A</sup>T<sub>E</sub>X 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.

**\marginnotetextwidth** 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** **\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 Issues Using `marginnote`

From version 1.3 `marginnote` does not longer support  $\text{\TeX}$  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  $\text{\TeX}$  engine without these primitives. Also  $\varepsilon\text{-}\text{\TeX}$  primitives are needed. However, with current free  $\text{\TeX}$  distributions like  $\text{\MiKTeX}$  or  $\text{\TeX Live}$  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.  $\text{\LaTeX}$  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  $\text{\LaTeX}$  run but only at the next. So after adding or deleting a margin note or after each change of page break you need two  $\text{\LaTeX}$  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  $\text{\XeTeX}$ ,  $\text{\PDFLaTeX}$  since version 1.40 or  $\text{\PDFLaTeX}$  before version 1.40 with PDF output and the horizontal position of a margin note is wrong, do one more  $\text{\PDFLaTeX}$  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`.

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 columns, 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

`\mn@savepos` 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  $\varepsilon$ -TeX extensions. So we do not longer need an explicit  $\varepsilon$ -TeX test.

```

1 \begingroup
2   \ifundefined{pdfsavepos}{%
3     \ifundefined{savepos}{%
4       \PackageError{marginnote}{%
5         neither \string\pdfsavepos\space nor \string\savepos\space
6         available
7       }{%
8         Package 'marginnote' depends on extended features of
9         PDFLaTeX, \MessageBreak
10        LuaLaTeX or XeLaTeX. It does not work without those
11        feature. \MessageBreak
12        If you'd continue the package will not provide any feature.
13      }%
14    \aftergroup\endinput
15  }{%
16    \ifundefined{lastxpos}{%
17      \PackageError{marginnote}{%
18        \string\savepos\space but not \string\lastxpos\space
19        available
20      }{%
21        Package 'marginnote' depends on extended features of
22        PDFLaTeX, \MessageBreak
23        LuaLaTeX or XeLaTeX. It does not work without those
24        feature. \MessageBreak
25        If you'd continue the package will not provide any feature.
26      }%
27    \aftergroup\endinput
28  }{%
29    \global\let\mn@savepos\savepos
30    \global\let\mn@lastxpos\lastxpos
31    \global\let\mn@pagewidth\pagewidth
32  }%
33 }
```

```

34 }{%
35   \@ifundefined{pdflastxpos}{%
36     \PackageError{marginnote}{%
37       \string\pdfsavepos\space but not \string\pdflastxpos\space
38       available
39     }{%
40       Package 'marginnote' depends on extended features of
41       PDFLaTeX, \MessageBreak
42       LuaLaTeX or XeLaTeX. It does not work without those
43       feature.\MessageBreak
44       If you'd continue the package will not provide any feature.
45     }%
46     \aftergroup\endinput
47   }{%
48     \global\let\mn@savepos\pdfsavepos
49     \global\let\mn@lastxpos\pdflastxpos
50     \global\let\mn@pagewidth\pdfpagewidth
51   }%
52 }%
53 \endgroup

```

Next declare and process the options.

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

```

54 \newif\ifmn@verbose\mn@verbosetrue
55 \DeclareOption{verbose}{\mn@verbosetrue}
56 \DeclareOption{quiet}{\mn@verbosefalse}

```

`\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`.

```

57 \newcommand*{\mn@strut}{\strut}

```

`\mn@zbox` This macro is used to set a horizontal box without height, depth and width.

```

58 \newcommand{\mn@zbox}[1]{}

```

The options do redefine both, `\mn@strut` and `\mn@zbox`.

```

59 \DeclareOption{fulladjust}{%
60   \renewcommand*{\mn@strut}{\strut}%
61   \renewcommand{\mn@zbox}[1]{%
62     \bgroup
63     \setbox\@tempboxa\vbox{#1}%
64     \ht\@tempboxa\ht\strutbox
65     \dp\@tempboxa\dp\strutbox
66     \wd\@tempboxa\z@
67     \box\@tempboxa
68   \egroup
69 }%
70 }

```

```

71 \DeclareOption{heightadjust}{%
72   \renewcommand*{\mn@strut}{\begingroup\dp\strutbox\z@\strut\endgroup}%
73   \renewcommand{\mn@zbox}[1]{%
74     \bgroup
75     \setbox\@tempboxa\vbox{#1}%
76     \ht\@tempboxa\ht\strutbox
77     \dp\@tempboxa\dp\z@
78     \wd\@tempboxa\z@
79     \box\@tempboxa
80   \egroup
81   }%
82 }
83 \DeclareOption{depthadjust}{%
84   \renewcommand*{\mn@strut}{\begingroup\ht\strutbox\z@\strut\endgroup}%
85   \renewcommand{\mn@zbox}[1]{%
86     \bgroup
87     \setbox\@tempboxa\vbox{#1}%
88     \ht\@tempboxa\ht\z@
89     \dp\@tempboxa\dp\strutbox
90     \wd\@tempboxa\z@
91     \box\@tempboxa
92   \egroup
93   }%
94 }
95 \DeclareOption{noadjust}{%
96   \renewcommand*{\mn@strut}{\relax}%
97   \renewcommand{\mn@zbox}[1]{%
98     \bgroup
99     \setbox\@tempboxa\vbox{\kern-\ht\strutbox #1}%
100    \ht\@tempboxa\ht\z@
101    \dp\@tempboxa\dp\z@
102    \wd\@tempboxa\z@
103    \box\@tempboxa
104  \egroup
105  }%
106 }

107 \ExecuteOptions{verbose,fulladjust}
108 \ProcessOptions\relax

```

`\newmarginnote` We need a macro to define a new note at the aux file. This will be done using the mechanism of L<sup>A</sup>T<sub>E</sub>X 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.

```

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

```

`\if@mn@pdfmode`  
`\@mn@mode@prefix`  
`\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 useful if the user can set it up. Because of this it is a user command.

```
110 \newcommand*{\marginnotetextwidth}{%
111 \let\marginnotetextwidth\textwidth
112 \AtBeginDocument{\edef\marginnotetextwidth{\the\textwidth}}
```

`\@mn@margintest` Macro `\@mn@margintest` does the complete test, which margin to use. The result may be found at `\if@tempwa`. 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` 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 `mn@abspage` at every `\@outputpage`. `\@mn@currpage` is the page from the page label if found. `\@mn@currxpos` is the real  $x$  position may be written with the page label and used to calculate the correct horizontal offset.

```
113 \newcommand*{\@mn@thispage}{%
114 \newcommand*{\@mn@currpage}{%
115 \newcommand*{\@mn@currxpos}{%
116 \newcounter{mn@abspage}
117 \AtBeginDocument{\setcounter{mn@abspage}{1}}%
118 \g@addto@macro\@outputpage{\stepcounter{mn@abspage}}%
119 \newcommand*{\@mn@margintest}{%
```

Number of the next margin note at this page.

```
120 \expandafter\ifx\csname @mn@thispage\endcsname\@empty
121 \gdef\@mn@atthispage{1}%
122 \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
123 \begingroup
124 \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
125 \xdef\@mn@atthispage{\the\@tempcnta}%
126 \endgroup
127 \else
128 \gdef\@mn@atthispage{1}%
129 \fi
130 \fi
131 \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.

```
132 \let\@mn@currpage\relax
133 \let\@mn@currxpos\relax
134 \mn@savepos
135 \protected@write\@auxout{\let\themn@abspage\relax}{%
136 \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
137 {\themn@abspage}{\noexpand\number\mn@lastxpos sp}}%
138 }%
```

If the margin note label was not defined, it seems to be new. In this case the absolute page number will be used for the test instead of the saved real page number.

```
139 \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.

```
140 \if@twoside
141 \if@mn@verbose
142 \PackageInfo{marginnote}{Suggest that margin
143 note \@mn@thispage.\@mn@atthispage\space will be on\MessageBreak
144 absolute page \themn@abspage.\MessageBreak
145 This may be wrong}%
146 \fi
147 \ifodd\value{mn@abspage}\@tempwattrue\else\@tempwafalse\fi
148 \else
149 \if@mn@verbose
150 \PackageInfo{marginnote}{right page because not two side mode}%
151 \fi
152 \@tempwattrue
153 \fi
154 \else
155 \edef\@mn@currpage{\csname
156 mn@note.\@mn@thispage.\@mn@atthispage\endcsname}%
157 \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. If `\@mn@currxpos` is not empty here, it should be corrected by `\hoffset` and maybe by `\pdfhorigin`.

```
158 \ifx\@mn@currxpos\@empty\else
159 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\hoffset\relax}%
160 \begingroup\expandafter\expandafter\expandafter\endgroup
161 \expandafter\ifx\csname pdfhorigin\endcsname\relax\else
162 \begingroup\expandafter\expandafter\expandafter\endgroup
163 \expandafter\ifx\csname pdfoutput\endcsname\relax
164 \begingroup\expandafter\expandafter\expandafter\endgroup
165 \expandafter\ifx\csname outputmode\endcsname\relax\else
166 \ifnum \outputmode=1 %
167 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
168 +1in\relax}%
169 \fi
170 \fi
171 \else
172 \ifnum \pdfoutput=1 %
173 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
174 +1in\relax}%
175 \fi
176 \fi
177 \fi
```

If you are using package `bidi` and RTL mode is active, the position is from right instead of left. So we have to subtract `\@mn@currxpos` from `\pdfpagewidth`



(or `\pagewidth` using LuaTeX, but this cannot be, because `bidi` is not LuaTeX-compatible).

```

178     \ifdefined\mn@pagewidth
179     \mn@if@RTL{%
180         \PackageInfo{marginnote}{Margin note
181             \@mn@thispage.\@mn@atthispage\space in RTL mode}%
182         \edef\mn@curr xpos{%
183             \the\dimexpr\mn@pagewidth-\mn@curr xpos\relax
184         }%
185     }{%
186     \fi
187 \fi
188 \edef\mn@currpage{\expandafter\@firstoftwo\mn@currpage}%
189 \if@mn@verbose
190     \PackageInfo{marginnote}{Margin note \@mn@thispage.\@mn@atthispage\space
191         is on absolute page \@mn@currpage}%
192 \fi
193 \if@twoside
194     \ifodd\mn@currpage\relax
195         \@tempswatrue
196         \if@twocolumn
197             \ifdim \mn@curr xpos
198                 < \dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
199                 \@tempswafalse
200             \fi
201         \fi
202     \else
203         \@tempswafalse
204         \if@twocolumn
205             \ifdim\mn@curr xpos>\dimexpr\evensidemargin+\columnwidth\relax
206                 \@tempswatrue
207             \fi
208         \fi
209     \fi
210 \else
211     \if@mn@verbose
212         \PackageInfo{marginnote}{right page because not two side mode}%
213     \fi
214     \@tempswatrue
215     \if@twocolumn
216         \ifdim \mn@curr xpos
217             < \dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
218             \@tempswafalse
219         \fi
220     \fi
221 \fi
222 \fi
223 }
```

```

@mn@ifRTL Test, whether or not \if@RTL exists and is true or false.
224 \newcommand*{\@mn@if@RTL}{%
225   \begingroup\expandafter\expandafter\expandafter\endgroup
226   \expandafter\ifx\csname if@RTL\endcsname\iftrue
227     \expandafter\@firstoftwo
228   \else
229     \expandafter\@secondoftwo
230   \fi
231 }

\marginnote Command \marginnote is the main macro of the package. The others are helpers
\@mn@marginnote to manage the optional arguments.
\@mn@@marginnote 232 \newcommand*{\marginnote}{%
\@mn@@@marginnote 233   \@dblarg\@mn@marginnote
234 }
235 \newcommand{\@mn@marginnote}[2][]{%
236   \ifhmode
237     \@bsphack
238     \begingroup
239     \ifdim\@savsk>\z@\else
240       \def\:{\@xifnch}\expandafter\def\:{ \futurelet\@let@token\@ifnch}%
241     \fi
242   \else
243     \begingroup
244     \fi
245   \@ifnextchar [{\@mn@@marginnote[#1][#2]}\@mn@@marginnote[#1][#2][\z@]}%
246 }
247 \newcommand{\@mn@@marginnote}{}
248 \long\def\@mn@@marginnote[#1][#2][#3]{%
249   \endgroup

In horizontal mode the space hack of the LATEX kernel will be used. In vertical
mode this should not be used.
250   \ifhmode
251     \@mn@@marginnote[#1][#2][#3]%
252     \@esphack
253   \else
254     \@mn@@marginnote[#1][#2][#3]%
255   \fi
256 }
257 \newcommand{\@mn@@@marginnote}{}
258 \long\def\@mn@@@marginnote[#1][#2][#3]{%

```

All changes (but change of counters that are global because of using the L<sup>A</sup>T<sub>E</sub>X 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 too 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.

```

259 \beginngroup
260   \ifmmode\mn@strut\let\@tempa\mn@vadjust\else
261     \if@inlabel\leavevmode\fi
262     \ifhmode\mn@strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
263   \fi
264   \@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 placed to the margin.

```

265     \vbox to\z@{%
266       \vss
267       \@mn@margintest
268       \if@reversemargin\if@tempswa
269         \@tempswafalse
270       \else
271         \@tempswatrue
272       \fi\fi
273       \if@tempswa
274         \rlap{%

```

If `\@mn@curr xpos` is neither `\relax` nor empty it is the real current  $x$  position of the last PDF<sup>1</sup>TeX run and may be used to calculate the real horizontal offset.

```

275         \ifmn@verbose
276         \PackageInfo{marginnote}{xpos seems to be \@mn@curr xpos}%
277       \fi
278     \beginngroup
279       \ifx\@mn@curr xpos\relax\else\ifx\@mn@curr xpos\empty\else
280         \kern-\dimexpr\@mn@curr xpos\relax
281       \fi\fi
282       \ifx\@mn@curr page\relax
283         \let\@mn@curr page\@ne
284       \fi
285       \if@twoside\ifodd\@mn@curr page\relax
286         \kern\oddsidemargin
287       \else
288         \kern\evensidemargin
289       \fi
290     \else
291       \kern\oddsidemargin
292     \fi
293     \kern 1in
294   \endgroup
295   \kern\marginnotetextwidth\kern\marginparsep
296   \vbox to\z@{\kern\marginnotevadjust\kern #3
297     \vbox to\z@{%
298       \hsize\marginparwidth
299
300       \linewidth\hsize

```

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

```

300          \kern-\parskip
301          \marginfont\raggedrightmarginnote\strut\hspace{\z@}%
302          \ignorespaces#2\endgraf
303          \vss}%
304      \vss}%
305  }%
306  \else
    Using the left margin.
307      \llap{%
308          \vbox to \z@{\kern\marginnotevadjust\kern #3
309          \vbox to \z@{%
310              \hsize\marginparwidth
311              \linewidth\hsize
    Same like above for left margins.
312          \kern-\parskip
313          \marginfont\raggedleftmarginnote\strut\hspace{\z@}%
314          \ignorespaces#1\endgraf
315          \vss
316          }%
317          \vss
318      }%
319      \if@mn@verbose
320          \PackageInfo{marginnote}{xpos seems to be \@mn@currxpos}%
321      \fi
322      \begingroup
323          \ifx\@mn@currxpos\relax\else\ifx\@mn@currpos\@empty\else
324              \kern\@mn@currxpos
325          \fi\fi
326          \ifx\@mn@currpage\relax
327              \let\@mn@currpage\@ne
328          \fi
329          \if@twoside\ifodd\@mn@currpage\relax
330              \kern-\oddsidemargin
331          \else
332              \kern-\evensidemargin
333          \fi
334          \else
335              \kern-\oddsidemargin
336          \fi
337          \kern-1in
338      \endgroup
339      \kern\marginparsep
340  }%
341  \fi
342  }%
343  }%
344  \endgroup
345 }

```

```

\marginnoterightadjust
\marginnoteleftadjust
\marginnotevadjust This may be used to define an automatic vertical adjust. The default tis zero.
                    Values greater than zero will move the margin note down, values less than zero
                    will move the margin note up.
346 \newcommand*{\marginnotevadjust}{%
347 \let\marginnotevadjust\z@

\mn@vlap This macro is used to set a vertical box without size at vertical mode.
348 \newcommand{\mn@vlap}[1]{%
349 \setbox\@tempboxa\vbox to \ht\strutbox{#1\vss}%
350 \box\@tempboxa\vskip-\baselineskip
351 }

\mn@vadjust This macro is used to set a vertical box at horizontal mode.
352 \newcommand{\mn@vadjust}[1]{%
353 \mn@zbox{\kern-\parskip
354 \leavevmode\vadjust{#1}%
355 \kern\parskip
356 }%
357 }

\marginfont These are very simple. A class may also define \marginfont. Use this if available.
\raggedleftmarginnote I don't use \let for the definitions of the ragged macros, so the meaning may
\raggedrightmarginnote change loading e.g. package ragged2e.
358 \providecommand*{\marginfont}{%
359 \newcommand*{\raggedleftmarginnote}{\raggedleft}
360 \newcommand*{\raggedrightmarginnote}{\raggedright}

```

## 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.

Symbols	I	
<code>\@mn@@@marginnote</code> . <u>232</u>	<code>\if@mn@pdfmode</code> .... <u>110</u>	<code>\marginnotevadjust</code> . ..... 2, <u>346</u>
<code>\@mn@@marginnote</code> .. <u>232</u>	<code>\if@mn@verbose</code> .... <u>54</u>	<code>\mn@abspage</code> ..... <u>113</u>
<code>\@mn@atthispage</code> ... <u>113</u>	M	<code>\mn@lastxpos</code> ..... <u>1</u>
<code>\@mn@currpage</code> ..... <u>113</u>	<code>\marginfont</code> ..... 2, <u>358</u>	<code>\mn@savepos</code> ..... <u>1</u>
<code>\@mn@currxpos</code> ..... <u>113</u>	<code>\marginnote</code> ..... 2, <u>232</u>	<code>\mn@strut</code> ..... <u>57</u>
<code>\@mn@ifRTL</code> ..... <u>224</u>	<code>\marginnoteleftadjust</code> ..... <u>346</u>	<code>\mn@vadjust</code> ..... <u>352</u>
<code>\@mn@marginnote</code> ... <u>232</u>	<code>\marginnoterightadjust</code> ..... <u>346</u>	<code>\mn@vlap</code> ..... <u>348</u>
<code>\@mn@margintest</code> ... <u>113</u>	<code>\marginnotetextwidth</code> ..... 2, <u>110</u>	<code>\mn@zbox</code> ..... <u>58</u>
<code>\@mn@mode@prefix</code> .. <u>110</u>	N	
<code>\@mn@thispage</code> ..... <u>113</u>	<code>\newmarginnote</code> .... <u>109</u>	

<b>R</b>	..... 2, <u>358</u>	..... 2, <u>358</u>
<code>\raggedleftmarginnote</code>	<code>\raggedrightmarginnote</code>	

## Change History

v1.0a	General: new options <code>fulladjust</code> , <code>heightadjust</code> , <code>depthadjust</code> , and <code>noadjust</code> ..... 5
<code>\marginfont</code> : Use	
<code>\providecommand</code> to define it. 13	
General: Example to macros	v1.1f
<code>\raggedleftmarginnote</code> and	<code>\@mn@@marginnote</code> : missing usage
<code>\raggedrightmarginnote</code> at	of <code>\marginnotevadjust</code> on left
documentation fixed [thanks to	margin fixed ..... 12
Susumu Tanimura]. ..... 2	v1.1g
v1.0b	<code>\@mn@@marginnote</code> : missing <code>\long</code>
General: spelling fixes ..... 1	added ..... 10
v1.1	set <code>\linewidth</code> ..... 11, 12
<code>\@mn@@marginnote</code> : new PDF	<code>\@mn@marginnote</code> : missing <code>\long</code>
mode feature ..... 10	added ..... 10
<code>\@mn@currpage</code> : new (internal) ... 7	v1.1i
<code>\@mn@currxpos</code> : new (internal) ... 7	<code>\@mn@@marginnote</code> :
<code>\@mn@margintest</code> : new PDF mode	<code>\ignorespaces</code> added ..... 11
feature ..... 7	<code>\strut</code> moved to fix
<code>\if@mn@pdfmode</code> : new switch .... 6	hyphenation (thanks to Ulrike
<code>\marginnotetextwidth</code> : new	Fischer) ..... 11
macro ..... 6	v1.2
v1.1a	<code>\@mn@mode@prefix</code> : (new (internal) 6
<code>\if@mn@pdfmode</code> : PDF <sub>T</sub> <sub>E</sub> <sub>X</sub> since	<code>\if@mn@pdfmode</code> : addition for
1.40 allows <code>\pdfsavepos</code> in	lua <sub>T</sub> <sub>E</sub> <sub>X</sub> from 0.85 ..... 6
DVI mode too ..... 6	<code>\mn@abspage</code> : addition for lua <sub>T</sub> <sub>E</sub> <sub>X</sub>
v1.1b	from 0.85 ..... 7
<code>\@mn@@marginnote</code> : use	v1.2a
<code>\mn@vadjust</code> instead of	<code>\mn@abspage</code> : redefine
<code>\vadjust</code> ..... 10	<code>\@mn@currxpos</code> only if not
<code>\if@mn@pdfmode</code> : if level fixed .... 6	empty ..... 8
<code>\mn@vadjust</code> : new (internal) .... 13	v1.2b
<code>\mn@zbox</code> : new (internal) ..... 5	<code>@mn@ifRTL</code> : new internal ..... 9
v1.1c	<code>\mn@abspage</code> : bidi code added .... 8
<code>\if@mn@pdfmode</code> :	General: spelling fixes (by Thomas
X <sub>Y</sub> <sub>T</sub> <sub>E</sub> <sub>X</sub> has working <code>\pdflastxpos</code> 6	Reuben) ..... 1
v1.1d	v1.3
<code>\mn@abspage</code> : take care of	<code>\@mn@mode@prefix</code> : removed ..... 6
<code>\hoffset</code> ..... 8	<code>\if@mn@pdfmode</code> : removed ..... 6
take care of <code>\pdfhorigin</code> ..... 8	<code>\marginnoteleftadjust</code> : removed 13
v1.1e	<code>\marginnoterightadjust</code> :
<code>\@mn@@marginnote</code> : use	removed ..... 13
<code>\mn@strut</code> instead of <code>\strut</code> . 10	<code>\mn@abspage</code> : twocolumn test
<code>\mn@strut</code> : new (semi internal) .. 5	added ..... 8
	non PDF mode removed ..... 7

use new internals <code>\mn@savepos</code>	command . . . . .	4
and <code>\mn@lastxpos</code> . . . . .		7
<code>\mn@lastxpos</code> : new internal	General: $\varepsilon$ -TeX removed . . . . .	4
command . . . . .	early <code>\pdfsavepos/\savepos</code>	
<code>\mn@savepos</code> : new internal	test . . . . .	4