

Non-Floating Margin Notes with `marginnote` Package*

Markus Kohm[†]

2018/07/01 v1.4

Abstract

In L^AT_EX 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 <code>marginnote</code> Package	1
2	Known Issues Using <code>marginnote</code>	3
3	Implementation	4

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.

[†]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 `\@parboxrestore` to restore the definition of `\par` and `\` and several other commands and sets `\parindent` and `\parskip` to 0, `\parfillskip` to 0pt plus 1fil and `\lineskip` to `\normallineskip` and `\baselineskip` to `\normalbaselineskip` for every margin note. This is the default since release 1.4 of `marginnote`.

`noparboxrestore` does not use `\@parboxrestore`.

`\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 `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.

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

`\raggedrightmarginnote`

- 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 \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 ε - \TeX primitives are needed. However, with current free \TeX distributions like MiK \TeX or \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. \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 X_YTeX, PDF_YTeX since version 1.40 or PDF_YTeX before version 1.40 with PDF output and the horizontal position of a margin note is wrong, do one more PDF_YTeX 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`
`\mn@lastxpos` or `\savepos` and `\lastxpos` and does not longer support engines without these primitives. All these engines also provide ε -TeX extensions. So we do not longer need an explicite ε -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         PDFYTeX, \MessageBreak
10        LuaYTeX or XeYTeX. 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        PDFYTeX, \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{pdfastxpos}{%
36         \PackageError{marginnote}{%
37             \string\pdfsavepos\space but not \string\pdfastxpos\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\pdfastxpos
50     \global\let\mn@pagewidth\pdfpagewidth
51 }%
52 }%
53 \endgroup

```

Next declare and process the options.

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

```

54 \newif\if@mn@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 }

```

`\mn@parboxrestore` We can either use `\@parboxrestore` inside the margin notes or dont use it. I

would recommend to use it, so this will be the new default.

```

107 \newcommand*{\mn@parboxrestore}{}
108 \DeclareOption{parboxrestore}{%
109   \renewcommand*{\mn@parboxrestore}{\@parboxrestore}%
110 }
111 \DeclareOption{noparboxrestore}{%
112   \renewcommand*{\mn@parboxrestore}{}%
113 }

```

```

114 \ExecuteOptions{verbose,fulladjust,parboxrestore}
115 \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^AT_EX 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.

```

116 \newcommand*{\newmarginnote}{\@newlabel{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.

```

117 \newcommand*{\marginnotetextwidth}{}
118 \let\marginnotetextwidth\textwidth
119 \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.

```

120 \newcommand*{\@mn@thispage}{}
121 \newcommand*{\@mn@currpage}{}
122 \newcommand*{\@mn@currxpos}{}
123 \newcounter{mn@abspage}
124 \AtBeginDocument{\setcounter{mn@abspage}{1}}%
125 \g@addto@macro\@outputpage{\stepcounter{mn@abspage}}%
126 \newcommand*{\@mn@margintest}{%

```

Number of the next margin note at this page.

```

127 \expandafter\ifx\csname @mn@thispage\endcsname\@empty
128   \gdef\@mn@atthispage{1}%

```

```

129 \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
130     \begingroup
131         \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
132         \xdef\@mn@atthispage{\the\@tempcnta}%
133     \endgroup
134 \else
135     \gdef\@mn@atthispage{1}%
136 \fi
137 \fi
138 \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.

```

139 \let\@mn@currpage\relax
140 \let\@mn@currxpos\relax
141 \mn@savapos
142 \protected@write\@auxout{\let\themn@abspage\relax}{%
143     \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
144         {\themn@abspage}{\noexpand\number\mn@lastxpos sp}}%
145 }%

```

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.

```

146 \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.

```

147 \if@twoside
148     \if@mn@verbose
149         \PackageInfo{marginnote}{Suggest that margin
150             note \@mn@thispage.\@mn@atthispage\space will be on\MessageBreak
151             absolute page \themn@abspage.\MessageBreak
152             This may be wrong}%
153     \fi
154     \ifodd\value{mn@abspage}\@tempwattrue\else\@tempwafalse\fi
155 \else
156     \if@mn@verbose
157         \PackageInfo{marginnote}{right page because not two side mode}%
158     \fi
159     \@tempwattrue
160 \fi
161 \else
162     \edef\@mn@currpage{\csname
163         mn@note.\@mn@thispage.\@mn@atthispage\endcsname}%
164     \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`.

```

165 \ifx\@mn@currxpos\empty\else

```



```

166 \edef\@mn@curr xpos{\the\dimexpr \@mn@curr xpos -\hoffset\relax}%
167 \begingroup\expandafter\expandafter\expandafter\endgroup
168 \expandafter\ifx\csname pdfhorigin\endcsname\relax\else
169 \begingroup\expandafter\expandafter\expandafter\endgroup
170 \expandafter\ifx\csname pdfoutput\endcsname\relax
171 \begingroup\expandafter\expandafter\expandafter\endgroup
172 \expandafter\ifx\csname outputmode\endcsname\relax\else
173 \ifnum \outputmode=1 %
174 \edef\@mn@curr xpos{\the\dimexpr \@mn@curr xpos -\pdfhorigin
175 +1in\relax}%
176 \fi
177 \fi
178 \else
179 \ifnum \pdfoutput=1 %
180 \edef\@mn@curr xpos{\the\dimexpr \@mn@curr xpos -\pdfhorigin
181 +1in\relax}%
182 \fi
183 \fi
184 \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@curr xpos` from `\pdfpagewidth` (or `\pagewidth` using `LuaTeX`, but this cannot be, because `bidi` is not `LuaTeX`-compatible).

```

185 \ifdefined\mn@pagewidth
186 \if@RTL{%
187 \PackageInfo{marginnote}{Margin note
188 \@mn@thispage.\@mn@atthispage\space in RTL mode}%
189 \edef\@mn@curr xpos{%
190 \the\dimexpr\mn@pagewidth-\@mn@curr xpos\relax
191 }%
192 }{}%
193 \fi
194 \fi
195 \edef\@mn@currpage{\expandafter\@firstoftwo\@mn@currpage}%
196 \if@mn@verbose
197 \PackageInfo{marginnote}{Margin note \@mn@thispage.\@mn@atthispage\space
198 is on absolute page \@mn@currpage}%
199 \fi
200 \if@twoside
201 \ifodd\@mn@currpage\relax
202 \@tempswatrue
203 \if@twocolumn
204 \ifdim \@mn@curr xpos
205 < \dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
206 \@tempswafalse
207 \fi
208 \fi
209 \else
210 \@tempswafalse

```

```

211         \if@twocolumn
212         \ifdim\@mn@currxpos>\dimexpr\evensidemargin+\columnwidth\relax
213         \@tempwattrue
214         \fi
215     \fi
216 \fi
217 \else
218     \if@mn@verbose
219     \PackageInfo{marginnote}{right page because not two side mode}%
220     \fi
221     \@tempwattrue
222     \if@twocolumn
223     \ifdim \@mn@currxpos
224         < \dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
225         \@tempwafalse
226     \fi
227 \fi
228 \fi
229 \fi
230 }

```

@mn@ifRTL Test, whether or not \if@RTL exists and is true or false.

```

231 \newcommand*{\@mn@if@RTL}{%
232     \begingroup\expandafter\expandafter\expandafter\endgroup
233     \expandafter\ifx\csname if@RTL\endcsname\iftrue
234     \expandafter\@firstoftwo
235 \else
236     \expandafter\@secondoftwo
237 \fi
238 }

```

\marginnote Command \marginnote is the main macro of the package. The others are helpers to manage the optional arguments.

```

\@mn@marginnote 239 \newcommand*{\marginnote}{%
\@mn@@marginnote 240     \@dblarg\@mn@marginnote
241 }
242 \newcommand{\@mn@marginnote}[2] [] {%
243     \ifhmode
244         \@bsphack
245         \begingroup
246         \ifdim\@savsk>\z@\else
247             \def\:{\@xifnch}\expandafter\def\:{ \futurelet\@let@token\@ifnch}%
248         \fi
249     \else
250         \begingroup
251     \fi
252     \@ifnextchar [{\@mn@marginnote[#1][#2]}{\@mn@marginnote[#1][#2][\z@]}%
253 }
254 \newcommand{\@mn@@marginnote}{%
255 \long\def\@mn@@marginnote[#1]#2[#3]{%

```

256 \endgroup

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

```

257 \ifhmode
258   \mn@@marginnote[{#1}]{#2}[{#3}]%
259   \esphack
260 \else
261   \mn@@marginnote[{#1}]{#2}[{#3}]%
262 \fi
263 }
264 \newcommand{\mn@@marginnote}{}
265 \long\def\mn@@marginnote[#1]#2[#3]{%
```

All changes (but change of counters that are global because of using the L^AT_EX 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.

```

266 \begingroup
267   \ifmmode\mn@strut\let\@tempa\mn@vadjust\else
268   \ifinlabel\leavevmode\fi
269   \ifhmode\mn@strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
270   \fi
271   \@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.

```

272   \vbox to\z@{
273     \vss
274     \mn@margintest
275     \if@reversemargin\if@tempswa
276       \@tempswafalse
277     \else
278       \@tempswatrue
279     \fi\fi
280     \if@tempswa
281       \rlap{%
```

If \mn@currxpos is neither \relax nor empty it is the real current x position of the last PDF^L_AT_EX run and may be used to calculate the real horizontal offset.

```

282     \ifmn@verbose
283     \PackageInfo{marginnote}{xpos seems to be \mn@currxpos}%
284     \fi
285     \begingroup
286     \ifx\mn@currxpos\relax\else\ifx\mn@currxpos\empty\else
287       \kern-\dimexpr\mn@currxpos\relax
```

```

288         \fi\fi
289         \ifx\@mn@currpage\relax
290             \let\@mn@currpage\@ne
291         \fi
292         \if@twoside\ifodd\@mn@currpage\relax
293             \kern\oddsidemargin
294         \else
295             \kern\evensidemargin
296         \fi
297     \else
298         \kern\oddsidemargin
299     \fi
300     \kern 1in
301 \endgroup
302 \kern\marginnotetextwidth\kern\marginparsep
303 \vbox to\z@{\kern\marginnotevadjust\kern #3
304     \vbox to\z@{%
305         \hsize\marginparwidth
306
307         \linewidth\hsize

```

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

```

307         \kern-\parskip
308         \mn@parboxrestore
309         \marginfont\raggedrightmarginnote\strut\hspace{\z@}%
310         \ignorespaces#2\endgraf
311         \vss}%
312     \vss}%
313 }%
314 \else

```

Using the left margin.

```

315     \llap{%
316         \vbox to\z@{\kern\marginnotevadjust\kern #3
317             \vbox to\z@{%
318                 \hsize\marginparwidth
319
320                 \linewidth\hsize

```

Same like above for left margins.

```

320         \kern-\parskip
321         \mn@parboxrestore
322         \marginfont\raggedleftmarginnote\strut\hspace{\z@}%
323         \ignorespaces#1\endgraf
324         \vss
325     }%
326     \vss
327 }%
328 \if@mn@verbose
329     \PackageInfo{marginnote}{xpos seems to be \@mn@currxpos}%
330 \fi

```

```

331         \begingroup
332         \ifx\@mn@currpos\relax\else\ifx\@mn@currpos\@empty\else
333             \kern\@mn@currpos
334         \fi\fi
335         \ifx\@mn@currpage\relax
336             \let\@mn@currpage\@ne
337         \fi
338         \if@twoside\ifodd\@mn@currpage\relax
339             \kern-\oddsidemargin
340         \else
341             \kern-\evensidemargin
342         \fi
343         \else
344             \kern-\oddsidemargin
345         \fi
346         \kern-1in
347     \endgroup
348     \kern\marginparsep
349 }%
350 \fi
351 }%
352 }%
353 \endgroup
354 }

```

`\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.

```
355 \newcommand*{\marginnotevadjust}{}

```

```
356 \let\marginnotevadjust\z@

```

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

```

357 \newcommand{\mn@vlap}[1]{%
358   \setbox\@tempboxa\vbox to \ht\strutbox{#1\vss}%
359   \box\@tempboxa\vskip-\baselineskip
360 }

```

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

```

361 \newcommand{\mn@vadjust}[1]{%
362   \mn@zbox{\kern-\parskip
363     \leavevmode\vadjust{#1}%
364     \kern\parskip
365   }%
366 }

```

`\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`.

```

367 \providecommand*{\marginfont}{}
368 \newcommand*{\raggedleftmarginnote}{\raggedleft}
369 \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	M	
\@mn@@marginnote . <u>239</u>	\marginfont 3, <u>367</u>	\mn@savepos <u>1</u>
\@mn@marginnote . . <u>239</u>	\marginnote 2, <u>239</u>	\mn@strut <u>57</u>
\@mn@atthispage . . . <u>120</u>	\marginnoteleftadjust <u>355</u>	\mn@vadjust <u>361</u>
\@mn@currpage <u>120</u>	\marginnoterightadjust <u>355</u>	\mn@vlap <u>357</u>
\@mn@currxpos <u>120</u>	\marginnotetextwidth 2, <u>117</u>	\mn@zbox <u>58</u>
\@mn@ifRTL <u>231</u>	\marginnotevadjust 2, <u>355</u>	
\@mn@marginnote . . . <u>239</u>	\mn@abspage <u>120</u>	N
\@mn@margintest . . . <u>120</u>	\mn@lastxpos <u>1</u>	\newmarginnote <u>116</u>
\@mn@mode@prefix . . <u>117</u>	\mn@parboxrestore . <u>107</u>	
\@mn@thispage <u>120</u>		R
		\raggedleftmarginnote 2, <u>367</u>
I		\raggedrightmarginnote 2, <u>367</u>
\if@mn@pdfmode <u>117</u>		
\if@mn@verbose <u>54</u>		

Change History

v1.0a	\marginfont: Use	\marginnotetextwidth: new macro 7
	\providecommand to define it. 13	
	General: Example to macros	v1.1a
	\raggedleftmarginnote and	\if@mn@pdfmode: PDF _T _E X since 1.40 allows \pdfsavepos in DVI mode too 7
	\raggedrightmarginnote at documentation fixed [thanks to Susumu Tanimura]. 2	v1.1b
		\@mn@@marginnote: use \mn@vadjust instead of \vadjust 11
v1.0b	General: spelling fixes 1	\if@mn@pdfmode: if level fixed 7
v1.1	\@mn@@marginnote: new PDF mode feature 10	\mn@vadjust: new (internal) 13
	\@mn@currpage: new (internal) . . . 7	\mn@zbox: new (internal) 5
	\@mn@currxpos: new (internal) . . . 7	v1.1c
	\@mn@margintest: new PDF mode feature 7	\if@mn@pdfmode: X _g _T _E X has working \pdflastxpos7
	\if@mn@pdfmode: new switch 7	v1.1d
		\mn@abspage: take care of

	\hoffset	8		\mn@currxpos only if not	
	take care of \pdfhorigin	8		empty	8
v1.1e			v1.2b		
	\mn@@@marginnote: use			\mn@ifRTL: new internal	10
	\mn@strut instead of \strut	11		\mn@abspage: bidi code added	9
	\mn@strut: new (semi internal)	5		General: spelling fixes (by Thomas	
	General: new options fulladjust,			Reuben)	1
	heightadjust, depthadjust,		v1.3		
	and noadjust	5		\mn@mode@prefix: removed	7
v1.1f				\ifmn@pdfmode: removed	7
	\mn@@@marginnote: missing usage			\marginnoteleftadjust: removed	13
	of \marginnotevadjust on left			\marginnoterightadjust:	
	margin fixed	12		removed	13
v1.1g				\mn@abspage: twocolumn test	
	\mn@@@marginnote: missing \long			added	9
	added	10		non PDF mode removed	8
	set \linewidth	12		use new internals \mn@savepos	
	\mn@@marginnote: missing \long			and \mn@lastxpos	8
	added	10		\mn@lastxpos: new internal	
v1.1i				command	4
	\mn@@@marginnote:			\mn@savepos: new internal	
	\ignorespaces added	12		command	4
	\strut moved to fix			General: ε -TeX removed	4
	hyphenation (thanks to Ulrike			early \pdfsavepos/\savepos	
	Fischer)	12		test	4
v1.2			v1.4		
	\mn@mode@prefix: (new (internal)	7		\mn@@@marginnote:	
	\ifmn@pdfmode: addition for			\mn@parboxrestore added	12
	luaTeX from 0.85	7		\mn@parboxrestore: new internal	
	\mn@abspage: addition for luaTeX			command	6
	from 0.85	7		General: new options	
v1.2a				parboxrestore (default) and	
	\mn@abspage: redefine			noparboxrestore	6