

Non-Floating Margin Notes with `marginnote` Package*

Markus Kohm[†]

2016/10/21 v1.2a

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 Problems 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 9, last revised 2016/10/21.

[†]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⟩]` 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^AT_EX 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 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 PDF_TE_X features (`\pdfsavepos` and `\pdfsavepos`). So if you are using PDF_LA_TE_X with PDF output or PDF_LA_TE_X with PDF_TE_X-version since 1.40 or X_EL_AT_EX you will not need to use the example code above, but you will need at least two L^AT_EX 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. those 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 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` somewhat different from `\marginpar` if used immediately 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 \XeTeX , \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 $\epsilon\text{-TeX}$.

```

1 \begingroup
2   \def\@tempb{}%
3   \def\@tempa{%
4     \PackageError{marginnote}{seems you are not running e-TeX\@tempb}{%
5       Since 2004 the LaTeX team recommends to use e-TeX.\MessageBreak
6       marginnote since version 1.1d uses e-TeX features.\MessageBreak
7       At actual systems 'latex' should already use e-TeX.\MessageBreak
8       At deprecated systems it may be called 'elatex'.\MessageBreak
9       Use either unsupported marginnote up to version 1.1c or.\MessageBreak
10      ask you administrator for LaTeX using e-TeX\@tempb.\MessageBreak
11      Not using e-TeX\@tempb\space is a fatal error!\MessageBreak
12      Processing cannot be continued!}%
13   \endgroup
14   \batchmode \errmessage{}\csname @@end\endcsname\end\relax
15   \csname endinput\endcsname
16 }%
17 \expandafter\ifx\csname eTeXversion\endcsname\relax\else
18   \ifnum\@eTeXversion < 2
19     \def\@tempb{ V 2}%
20   \else
21     \let\@tempa\endgroup
22   \fi
23 \fi
24 \@tempa

```

Next declare and process the options.

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

```

25 \newif\if@mn@verbose\@mn@verbosetrue

```

```

26 \DeclareOption{verbose}{\@mn@verbosetrue}
27 \DeclareOption{quiet}{\@mn@verbosefalse}

```

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

```

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

```

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

```

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

```

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

```

30 \DeclareOption{fulladjust}{%
31   \renewcommand*{\mn@strut}{\strut}%
32   \renewcommand{\mn@zbox}[1]{%
33     \bgroup
34       \setbox\@tempboxa\vbox{#1}%
35       \ht\@tempboxa\ht\strutbox
36       \dp\@tempboxa\dp\strutbox
37       \wd\@tempboxa\z@
38       \box\@tempboxa
39     \egroup
40   }%
41 }
42 \DeclareOption{heightadjust}{%
43   \renewcommand*{\mn@strut}{\begingroup\dp\strutbox\z@\strut\endgroup}%
44   \renewcommand{\mn@zbox}[1]{%
45     \bgroup
46       \setbox\@tempboxa\vbox{#1}%
47       \ht\@tempboxa\ht\strutbox
48       \dp\@tempboxa\dp\z@
49       \wd\@tempboxa\z@
50       \box\@tempboxa
51     \egroup
52   }%
53 }
54 \DeclareOption{depthadjust}{%
55   \renewcommand*{\mn@strut}{\begingroup\ht\strutbox\z@\strut\endgroup}%
56   \renewcommand{\mn@zbox}[1]{%
57     \bgroup
58       \setbox\@tempboxa\vbox{#1}%
59       \ht\@tempboxa\ht\z@
60       \dp\@tempboxa\dp\strutbox
61       \wd\@tempboxa\z@
62       \box\@tempboxa
63     \egroup
64   }%
65 }
66 \DeclareOption{noadjust}{%

```

```

67 \renewcommand*{\mn@strut}{\relax}%
68 \renewcommand{\mn@zbox}[1]{%
69   \bgroup
70   \setbox\@tempboxa\vbox{\kern-\ht\strutbox #1}%
71   \ht\@tempboxa\ht\z@
72   \dp\@tempboxa\dp\z@
73   \wd\@tempboxa\z@
74   \box\@tempboxa
75   \egroup
76 }%
77 }

78 \ExecuteOptions{verbose,fulladjust}
79 \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.

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

`\if@mn@pdfmode` We need to know, whether or not PDF_TE_X and which version of PDF_TE_X is used.
`\@mn@mode@prefix` With PDF_TE_X the horizontal output position may be detected using `\pdfsavepos` and `\pdflastxpos`. So the relative position of the margin may be calculated. Without PDF_TE_X only manual adjustment is available. While PDF mode or not may change before start of the document, setting up the switch is delayed.

```

81 \newif\if@mn@pdfmode\@mn@pdfmodefalse
82 \newcommand*{\@mn@mode@prefix}{pdf}
83 \AtBeginDocument{%
84   \begingroup\expandafter\expandafter\expandafter\endgroup
85   \expandafter\ifx\csname pdflastxpos\endcsname\relax
86     \begingroup\expandafter\expandafter\expandafter\endgroup
87     \expandafter\ifx\csname lastxpos\endcsname\relax\else
88       \@mn@pdfmodetrue
89       \renewcommand*{\@mn@mode@prefix}{}%
90     \fi
91   \else % bg or 1
92     \begingroup\expandafter\expandafter\expandafter\endgroup
93     \expandafter\ifx\csname pdftexversion\endcsname\relax % bg 2
94       \begingroup\expandafter\expandafter\expandafter\endgroup
95       \expandafter\ifx\csname pdfoutput\endcsname\relax % bg 3
96         \begingroup\expandafter\expandafter\expandafter\endgroup
97         \expandafter\ifx\csname XeTeXrevision\endcsname\relax\else % bg 4
98           \@mn@pdfmodetrue
99         \fi % ed 4
100       \else % or 3
101         \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi % bg ed 4
102       \fi % ed 3
103     \else % or 2

```

```

104     \ifnum \pdfTeXversion<140 % bg 3
105     \begingroup\expandafter\expandafter\expandafter\endgroup
106     \expandafter\ifx\csname pdfoutput\endcsname\relax % bg 4
107     \else % or 4
108     \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi % bg ed 5
109     \fi % ed 4
110     \else % or 3
111     \@mn@pdfmodetrue
112     \fi % ed 3
113     \fi % ed 2
114 \fi % ed 1
115 \if@mn@verbose
116     \if@mn@pdfmode
117     \PackageInfo{marginnote}{%
118     \string\pdfoutput\space not 0 or unimportant and\MessageBreak
119     \string\pdflastxpos\space or \string\lastxpos\space
120     available.\MessageBreak
121     Extended position detection mode activated\@gobble
122     }%
123     \else
124     \PackageInfo{marginnote}{%
125     either \string\pdflastxpos\space or \string\pdfoutput\space not
126     available\MessageBreak
127     or \string\pdfoutput\space set to 0.\MessageBreak
128     Extended position detection mode deactivated\@gobble
129     }%
130     \fi
131 \fi
132 }

```

`\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.

```

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

```

`\@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 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 PDF_TE_X 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!

```

136 \newcommand*{\@mn@thispage}{%
137 \newcommand*{\@mn@currpage}{%
138 \newcommand*{\@mn@currxpos}{%
139 \newcounter{mn@abspage}
140 \AtBeginDocument{\setcounter{mn@abspage}{1}%
141 \g@addto@macro\@outputpage{\stepcounter{mn@abspage}}}%
142 \newcommand*{\@mn@margintest}{%

```

Number of the next margin note at this page.

```

143 \expandafter\ifx\csname @mn@thispage\endcsname\@empty
144 \gdef\@mn@atthispage{1}%
145 \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
146 \begin{group}
147 \tempcnta \@mn@atthispage\advance\tempcnta by \@ne
148 \xdef\@mn@atthispage{\the\tempcnta}%
149 \end{group}
150 \else
151 \gdef\@mn@atthispage{1}%
152 \fi
153 \fi
154 \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.

```

155 \let\@mn@currpage\relax
156 \let\@mn@currxpos\relax
157 \if@mn@pdfmode
158 \nameuse{\@mn@mode@prefix savepos}%
159 \protected@write\@auxout{\let\themn@abspage\relax}{%
160 \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
161 {\themn@abspage}{\noexpand\number\@nameuse{\@mn@mode@prefix lastxpos}sp}}}%
162 }%
163 \else
164 \protected@write\@auxout{\let\themn@abspage\relax}{%
165 \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
166 {\themn@abspage}{}}}%
167 }%
168 \fi

```

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.

```

169 \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.

```

170 \if@twoside
171 \if@mn@verbose
172 \PackageInfo{marginnote}{Suggest that margin
173 note \@mn@thispage.\@mn@atthispage\space will be on\MessageBreak

```



```

174         absolute page \themn@abspage.\MessageBreak
175         This may be wrong}%
176     \fi
177     \ifodd\value{mn@abspage}\@tempwattrue\else\@tempwafalse\fi
178 \else
179     \if@mn@verbose
180         \PackageInfo{marginnote}{right page because not two side mode}%
181     \fi
182     \@tempwattrue
183 \fi
184 \else
185     \edef\@mn@currpage{\csname
186         mn@note.\@mn@thispage.\@mn@atthispage\endcsname}%
187     \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`.

```

188     \ifx\@mn@currxpos\empty\else
189         \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\hoffset\relax}%
190         \begingroup\expandafter\expandafter\expandafter\endgroup
191         \expandafter\ifx\csname pdfhorigin\endcsname\relax\else
192             \begingroup\expandafter\expandafter\expandafter\endgroup
193             \expandafter\ifx\csname pdfoutput\endcsname\relax
194                 \begingroup\expandafter\expandafter\expandafter\endgroup
195                 \expandafter\ifx\csname outputmode\endcsname\relax\else
196                     \ifnum \outputmode=1 %
197                         \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
198                             +1in\relax}%
199                     \fi
200                 \fi
201             \else
202                 \ifnum \pdfoutput=1 %
203                     \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
204                         +1in\relax}%
205                 \fi
206             \fi
207         \fi
208     \fi
209     \edef\@mn@currpage{\expandafter\@firstoftwo\@mn@currpage}%
210     \if@mn@verbose
211         \PackageInfo{marginnote}{Margin note \@mn@thispage.\@mn@atthispage\space
212             is on absolute page \@mn@currpage\MessageBreak}%
213     \fi
214     \if@twoside
215         \ifodd\@mn@currpage\relax
216             \@tempwattrue
217         \else
218             \@tempwafalse

```

```

219     \fi
220   \else
221     \if@mn@verbose
222       \PackageInfo{marginnote}{right page because not two side mode}%
223     \fi
224     \@tempwattrue
225   \fi
226 \fi
227 }

```

```

\marginnote Command \marginnote is the main macro of the package. The others are helpers
\@mn@marginnote to manage the optional arguments.
\@mn@@marginnote 228 \newcommand*{\marginnote}{%
\@mn@@@marginnote 229 \@dblarg\@mn@marginnote
230 }
231 \newcommand{\@mn@marginnote}[2] [] {%
232   \ifhmode
233     \@bsphack
234     \begingroup
235     \ifdim\@savsk>\z@ \else
236       \def\:{\@xifnch}\expandafter\def\:{ { \futurelet\@let@token\@ifnch}%
237     \fi
238   \else
239     \begingroup
240   \fi
241   \@ifnextchar [{\@mn@@marginnote[#1]{#2}}{\@mn@@marginnote[#1]{#2}[\z@]}%
242 }
243 \newcommand{\@mn@@marginnote}{%
244 \long\def\@mn@@marginnote[#1]#2[#3]{%
245   \endgroup

```

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

```

246   \ifhmode
247     \@mn@@@marginnote[#1]{#2}[\z@]{#3}%
248     \@esphack
249   \else
250     \@mn@@@marginnote[#1]{#2}[\z@]{#3}%
251   \fi
252 }
253 \newcommand{\@mn@@@marginnote}{%
254 \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.

```

255 \begingroup
256 \ifmmode\mn@strut\let\@tempa\mn@vadjust\else
257 \if@inlabel\leavevmode\fi
258 \ifhmode\mn@strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
259 \fi
260 \@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 switches. Then the note will be placed to the margin.

```

261 \vbox to\z@{%
262 \vss
263 \@mn@margintest
264 \if@reversemargin\if@tempswa
265 \@tempswafalse
266 \else
267 \@tempswatrue
268 \fi\fi
269 \if@tempswa
270 \rlap{%

```

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

```

271 \ifx\@mn@currxpos\relax
272 \kern\marginnoterightadjust
273 \if@mn@verbose
274 \PackageInfo{marginnote}{%
275 xpos not known,\MessageBreak
276 using \string\marginnoterightadjust}%
277 \fi
278 \else\ifx\@mn@currxpos\@empty
279 \kern\marginnoterightadjust
280 \if@mn@verbose
281 \PackageInfo{marginnote}{%
282 xpos not known,\MessageBreak
283 using \string\marginnoterightadjust}%
284 \fi
285 \else
286 \if@mn@verbose
287 \PackageInfo{marginnote}{%
288 xpos seems to be \@mn@currxpos,\MessageBreak
289 \string\marginnoterightadjust
290 \space ignored}%
291 \fi
292 \begingroup
293 \setlength{\@tempdima}{\@mn@currxpos}%
294 \kern-\@tempdima
295 \if@twoside\ifodd\@mn@currpage\relax
296 \kern\oddsidemargin
297 \else

```

```

298         \kern\evensidemargin
299     \fi
300     \else
301         \kern\oddsidemargin
302     \fi
303     \kern 1in
304 \endgroup
305 \fi
306 \fi
307 \kern\marginnotetextwidth\kern\marginparsep
308 \vbox to\z@{\kern\marginnotevadjust\kern #3
309     \vbox to\z@{%
310         \hsize\marginparwidth
311
312         \linewidth\hsize

```

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

```

312         \kern-\parskip
313         \marginfont\raggedrightmarginnote\strut\hspace{\z@}%
314         \ignorespaces#2\endgraf
315         \vss}%
316     \vss}%
317 }%
318 \else

```

Using the left margin.

```

319     \llap{%
320         \vbox to\z@{\kern\marginnotevadjust\kern #3
321             \vbox to\z@{%
322                 \hsize\marginparwidth
323
324                 \linewidth\hsize

```

Same like above for left margins.

```

324         \kern-\parskip
325         \marginfont\raggedleftmarginnote\strut\hspace{\z@}%
326         \ignorespaces#1\endgraf
327         \vss}%
328     \vss}%
329 \ifx\@mn@curr xpos\relax
330     \kern\marginnoteleftadjust
331     \if@mn@verbose
332         \PackageInfo{marginnote}{%
333             xpos not known,\MessageBreak
334             using \string\marginnoteleftadjust}%
335     \fi
336 \else\ifx\@mn@curr xpos\empty
337     \kern\marginnoteleftadjust
338     \if@mn@verbose
339         \PackageInfo{marginnote}{%
340             xpos not known,\MessageBreak

```

```

341         using \string\marginnoteleftadjust}%
342     \fi
343 \else
344     \if@mn@verbose
345         \PackageInfo{marginnote}{%
346             xpos seems to be \@mn@currxpos,\MessageBreak
347             \string\marginnoteleftadjust
348             \space ignored}%
349     \fi
350 \beginngroup
351     \kern\@mn@currxpos
352     \if@twoside\ifodd\@mn@currxpos\relax
353         \kern-\oddsidemargin
354     \else
355         \kern-\evensidemargin
356     \fi
357 \else
358     \kern-\oddsidemargin
359 \fi
360 \kern-1in
361 \endgroup
362 \fi
363 \fi
364 \kern\marginparsep
365 }%
366 \fi
367 }%
368 }%
369 \endgroup
370 }

```

`\marginnoterightadjust` These may be used to define an automatic horizontal adjust. The default is zero.

`\marginnoteleftadjust` They will be used only if not PDF_TEX or PDF_TEX before version 1.40 in DVI mode is used, because in this case the save position features are not available.

```

371 \newcommand*{\marginnoterightadjust}{}
372 \newcommand*{\marginnoteleftadjust}{}
373 \let\marginnoterightadjust\z@
374 \let\marginnoteleftadjust\z@

```

`\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.

```

375 \newcommand*{\marginnotevadjust}{}
376 \let\marginnotevadjust\z@

```

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

```

377 \newcommand{\mn@vlap}[1]{%
378     \setbox\@tempboxa\vbox to \ht\strutbox{#1\vss}%
379     \box\@tempboxa\vskip-\baselineskip
380 }

```

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

```

381 \newcommand{\mn@vadjust}[1]{%
382   \mn@zbox{\kern-\parskip
383     \leavevmode\vadjust{#1}%
384     \kern\parskip
385   }%
386 }
```

`\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`.

```

387 \providecommand*\marginfont{}
388 \newcommand*\raggedleftmarginnote{\raggedleft}
389 \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	<code>\if@mn@verbose</code> 25	<code>\mn@strut</code> 28
<code>\@mn@@marginnote</code> . 228		
<code>\@mn@marginnote</code> . . 228	M	<code>\mn@vadjust</code> 381
<code>\@mn@atthispage</code> . . . 136	<code>\marginfont</code> 3, 387	<code>\mn@vlap</code> 377
<code>\@mn@currpage</code> 136	<code>\marginnote</code> 2, 228	<code>\mn@zbox</code> 29
<code>\@mn@currpos</code> 136	<code>\marginnoteleftadjust</code>	
<code>\@mn@marginnote</code> . . . 228 2, 371	N
<code>\@mn@margintest</code> . . . 136	<code>\marginnoterightadjust</code>	<code>\newmarginnote</code> 80
<code>\@mn@mode@prefix</code> . . 81 2, 371	
<code>\@mn@thispage</code> 136	<code>\marginnotetextwidth</code>	R
 2, 133	<code>\raggedleftmarginnote</code>
I	<code>\marginnotevadjust</code> 3, 387
<code>\if@mn@pdfmode</code> 81 3, 375	<code>\raggedrightmarginnote</code>
	<code>\mn@abspage</code> 136 3, 387

Change History

v1.0a		v1.0b
<code>\marginfont</code> : Use		General: spelling fixes 1
<code>\providecommand</code> to define it. 14	v1.1	
General: Example to macros	<code>\@mn@@marginnote</code> : new PDF	
<code>\raggedleftmarginnote</code> and	mode feature 10	
<code>\raggedrightmarginnote</code> at	<code>\@mn@currpage</code> : new (internal) . . . 7	
documentation fixed [thanks to	<code>\@mn@currpos</code> : new (internal) . . . 7	
Susumu Tanimura]. 3	<code>\@mn@margintest</code> : new PDF mode	

feature	7	and noadjust	5
\if@mn@pdfmode: new switch	6	v1.1f	
\marginnotetextwidth: new macro	7	\@mn@@marginnote: missing usage of \marginnotevadjust on left margin fixed	12
v1.1a		v1.1g	
\if@mn@pdfmode: PDF _T _E X since 1.40 allows \pdfsavepos in DVI mode too	6	\@mn@@marginnote: missing \long added	10
v1.1b		set \linewidth	12
\@mn@@marginnote: use \mn@vadjust instead of \vadjust	10	\@mn@@marginnote: missing \long added	10
\if@mn@pdfmode: if level fixed	6	v1.1i	
\mn@vadjust: new (internal)	14	\@mn@@marginnote: \ignorespaces added	12
\mn@zbox: new (internal)	5	\strut moved to fix hyphenation (thanks to Ulrike Fischer)	12
v1.1c		v1.2	
\if@mn@pdfmode: X _T _E _X has working \pdflastxpos6		\@mn@mode@prefix: (new (internal)	6
v1.1d		\if@mn@pdfmode: addition for lua _T _E _X from 0.85	6
\mn@abspage: take care of \hoffset	9	\mn@abspage: addition for lua _T _E _X from 0.85	8
take care of \pdfhorigin	9	v1.2a	
v1.1e		\mn@abspage: redefine \mn@currxpos only if not empty	9
\@mn@@marginnote: use \mn@strut instead of \strut ..	10		
\mn@strut: new (semi internal) ..	5		
General: new options fulladjust, heightadjust, depthadjust,			