

Non-Floating Margin Notes with `marginnote` Package*

Markus Kohm[†]

2018/07/01 v1.4a

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 23, last revised 2018/07/21.

[†]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.

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

From version 1.3 `marginnote` does not longer support \TeX engines without primitives `\pdfsavepos/\savepos` and `\pdflastxpos/\lastxpos`. The former (manual adjustment) fallback has been removed. You'll get an error message, if you try to use a \TeX engine without these primitives. Also $\epsilon\text{-}\text{\TeX}$ 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 \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`.

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 `\pdfsavepos` 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         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.

```

\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}{}

```

`\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 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.
116 \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.
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
\@mn@thispage new note on the first page, try to count the notes by page. We know that this
\@mn@atthispage can not be successful, but never the less it may be a good try. \@mn@thispage
\@mn@currpage saves the page number of the last usage of \@mn@margintest. \@mn@atthispage
\@mn@currxpos saves the number of margin note at this page. But we need to know the absolut
\mn@abspage 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 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{%
126 \stepcounter{mn@abspage}%

```

From version 1.4a there is a workaround for consecutive odd pages or consecutive even pages in a twoside document.

```

127 \ifodd\value{mn@abspage}%
128 \ifodd\value{page}%
129 \else
130 \if@twoside
131 \begingroup
132 \advance\c@page\m@ne
133 \PackageWarningNoLine{marginnote}{%
134 Consecutive odd pages found.\MessageBreak
135 Note, it is not recommended to use consecutive\MessageBreak
136 odd pages in a double-ended document.\MessageBreak
137 The pages of your document should always\MessageBreak
138 be a sequence: odd-even-odd-even-...\MessageBreak
139 Maybe you've forgotten a
140 \@ifundefined{KOMAClassName}{%
141 {\string\cleardoublepage}%
142 {\string\cleardoubleoddpage}%
143 before\MessageBreak
144 changing the page numbering on page \thepage
145 }%
146 \endgroup
147 \fi
148 \PackageInfo{marginnote}{Using workaround for absolute page number}%
149 \stepcounter{mn@abspage}%
150 \fi
151 \else
152 \ifodd\value{page}%
153 \if@twoside
154 \begingroup
155 \advance\c@page\m@ne
156 \PackageWarningNoLine{marginnote}{%
157 Consecutive even pages found.\MessageBreak
158 Note, it is not recommended to use consecutive\MessageBreak
159 even pages in a double-ended document.\MessageBreak
160 The pages of your document should always\MessageBreak
161 be a sequence: odd-even-odd-even-...\MessageBreak
162 Maybe you've forgotten a
163 \@ifundefined{KOMAClassName}{%
164 {\string\cleardoublepage}%
165 {\string\cleardoubleevenpage}%
166 before\MessageBreak
167 changing the page numbering on page \thepage
168 }%
169 \endgroup
170 \fi
171 \PackageInfo{marginnote}{Using workaround for absolute page number}%
172 \stepcounter{mn@abspage}%
173 \fi

```



```

174     \fi
175   }%
176 }
177 \newcommand*{\@mn@margintest}{%
  Number of the next margin note at this page.
178   \expandafter\ifx\csname @mn@thispage\endcsname\@empty
179     \gdef\@mn@atthispage{1}%
180   \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
181     \begingroup
182       \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
183       \xdef\@mn@atthispage{\the\@tempcnta}%
184     \endgroup
185   \else
186     \gdef\@mn@atthispage{1}%
187   \fi
188 \fi
189 \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.

```

190 \let\@mn@currpage\relax
191 \let\@mn@currxpos\relax
192 \mn@savapos
193 \protected@write\@auxout{\let\themn@abspage\relax}{%
194   \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
195     {\themn@abspage}{\noexpand\number\mn@lastxpos sp}}%
196 }%

```

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.

```

197 \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.

```

198   \if@twoside
199     \if@mn@verbose
200       \PackageInfo{marginnote}{Suggest that margin
201         note \@mn@thispage.\@mn@atthispage\space will be on\MessageBreak
202         absolute page \themn@abspage.\MessageBreak
203         This may be wrong}%
204     \fi
205     \ifodd\value{mn@abspage}\@tempwattrue\else\@tempwafalse\fi
206   \else
207     \if@mn@verbose
208       \PackageInfo{marginnote}{right page because not two side mode}%
209     \fi
210     \@tempwattrue
211   \fi
212 \else
213   \edef\@mn@currpage{\csname

```

```

214     mn@note.\@mn@thispage.\@mn@atthispage\endcsname}%
215     \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`.

```

216     \ifx\@mn@currxpos\empty\else
217         \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\hoffset\relax}%
218         \begingroup\expandafter\expandafter\expandafter\endgroup
219         \expandafter\ifx\csname pdfhorigin\endcsname\relax\else
220             \begingroup\expandafter\expandafter\expandafter\endgroup
221             \expandafter\ifx\csname pdfoutput\endcsname\relax
222                 \begingroup\expandafter\expandafter\expandafter\endgroup
223                 \expandafter\ifx\csname outputmode\endcsname\relax\else
224                     \ifnum \outputmode=1 %
225                         \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
226                             +1in\relax}%
227                     \fi
228                 \fi
229             \else
230                 \ifnum \pdfoutput=1 %
231                     \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
232                         +1in\relax}%
233                 \fi
234             \fi
235         \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).

```

236     \ifdefined\mn@pagewidth
237         \mn@ifRTL{%
238             \PackageInfo{marginnote}{Margin note
239                 \@mn@thispage.\@mn@atthispage\space in RTL mode}%
240             \edef\@mn@currxpos{%
241                 \the\dimexpr\mn@pagewidth-\@mn@currxpos\relax
242             }%
243         }{}%
244     \fi
245 \fi
246 \edef\@mn@currpage{\expandafter\@firstoftwo\@mn@currpage}%
247 \if@mn@verbose
248     \PackageInfo{marginnote}{Margin note \@mn@thispage.\@mn@atthispage\space
249         is on absolute page \@mn@currpage}%
250 \fi
251 \if@twoside
252     \ifodd\@mn@currpage\relax
253         \@tempwatrue

```

```

254         \if@twocolumn
255         \ifdim \@mn@currxpos
256             < \dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
257             \@tempwafalse
258         \fi
259     \fi
260 \else
261     \@tempwafalse
262     \if@twocolumn
263         \ifdim \@mn@currxpos>\dimexpr\evensidemargin+\columnwidth\relax
264             \@tempwattrue
265         \fi
266     \fi
267 \fi
268 \else
269     \if@mn@verbose
270         \PackageInfo{marginnote}{right page because not two side mode}%
271     \fi
272     \@tempwattrue
273     \if@twocolumn
274         \ifdim \@mn@currxpos
275             < \dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
276             \@tempwafalse
277         \fi
278     \fi
279 \fi
280 \fi
281 }

```

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

```

282 \newcommand*{\@mn@if@RTL}{%
283     \begingroup\expandafter\expandafter\expandafter\endgroup
284     \expandafter\ifx\csname if@RTL\endcsname\iftrue
285         \expandafter\@firstoftwo
286     \else
287         \expandafter\@secondoftwo
288     \fi
289 }

```

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

```

\@mn@marginnote 290 \newcommand*{\marginnote}{%
\@mn@@marginnote 291     \@dblarg\@mn@marginnote
292 }
293 \newcommand{\@mn@marginnote}[2] [] {%
294     \ifhmode
295         \@bsphack
296         \begingroup
297         \ifdim\@savsk>\z@ \else
298             \def\:{\@xifnch}\expandafter\def\:{ { \futurelet\@let@token\@ifnch}%

```

```

299   \fi
300   \else
301     \begingroup
302     \fi
303     \ifnextchar [{\@mn@@marginnote[{#1}]{#2}}{\@mn@@marginnote[{#1}]{#2}[\z@]}%
304   }
305   \newcommand{\@mn@@marginnote}{}
306   \long\def\@mn@@marginnote[#1]#2[#3]{%
307     \endgroup

```

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

```

308   \ifhmode
309     \@mn@@marginnote[{#1}]{#2}[\z@]{#3}%
310     \@esphack
311   \else
312     \@mn@@marginnote[{#1}]{#2}[\z@]{#3}%
313   \fi
314 }
315 \newcommand{\@mn@@marginnote}{}
316 \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.

```

317   \begingroup
318     \ifmmode\mn@strut\let\@tempa\mn@vadjust\else
319       \if@inlabel\leavevmode\fi
320       \ifhmode\mn@strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
321     \fi
322     \@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.

```

323       \vbox to\z@{%
324         \vss
325         \@mn@margintest
326         \if@reversemargin\if@tempswa
327           \@tempswafalse
328         \else
329           \@tempswatrue
330         \fi\fi
331         \if@tempswa
332           \rlap{%

```

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

```

333         \if@mn@verbose
334         \PackageInfo{marginnote}{xpos seems to be \@mn@currxpos}%
335         \fi
336         \begingroup
337         \ifx\@mn@currxpos\relax\else\ifx\@mn@currxpos\@empty\else
338             \kern-\dimexpr\@mn@currxpos\relax
339         \fi\fi
340         \ifx\@mn@currpage\relax
341             \let\@mn@currpage\@ne
342         \fi
343         \if@twoside\ifodd\@mn@currpage\relax
344             \kern\oddsidemargin
345         \else
346             \kern\evensidemargin
347         \fi
348         \else
349             \kern\oddsidemargin
350         \fi
351         \kern 1in
352     \endgroup
353     \kern\marginnotetextwidth\kern\marginparsep
354     \vbox to\z@{\kern\marginnotevadjust\kern #3
355         \vbox to\z@{%
356             \hsize\marginparwidth
357
358             \linewidth\hsize

```

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

```

358         \kern-\parskip
359         \mn@parboxrestore
360         \marginfont\raggedrightmarginnote\strut\hspace{\z@}%
361         \ignorespaces#2\endgraf
362         \vss}%
363     \vss}%
364 }%
365 \else

```

Using the left margin.

```

366     \llap{%
367         \vbox to\z@{\kern\marginnotevadjust\kern #3
368             \vbox to\z@{%
369                 \hsize\marginparwidth
370
371                 \linewidth\hsize

```

Same like above for left margins.

```

371         \kern-\parskip
372         \mn@parboxrestore
373         \marginfont\raggedleftmarginnote\strut\hspace{\z@}%

```

```

374         \ignorespaces#1\endgraf
375         \vss
376     }%
377     \vss
378 }%
379 \if@mn@verbose
380     \PackageInfo{marginnote}{xpos seems to be \@mn@currxpos}%
381 \fi
382 \begingroup
383     \ifx\@mn@currxpos\relax\else\ifx\@mn@currpos\@empty\else
384         \kern\@mn@currxpos
385     \fi\fi
386     \ifx\@mn@currpage\relax
387         \let\@mn@currpage\@ne
388     \fi
389     \if@twoside\ifodd\@mn@currpage\relax
390         \kern-\oddsidemargin
391     \else
392         \kern-\evensidemargin
393     \fi
394 \else
395     \kern-\oddsidemargin
396 \fi
397 \kern-1in
398 \endgroup
399 \kern\marginparsep
400 }%
401 \fi
402 }%
403 }%
404 \endgroup
405 }

```

`\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.

```

406 \newcommand*{\marginnotevadjust}{}
407 \let\marginnotevadjust\z@

```

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

```

408 \newcommand{\mn@vlap}[1]{%
409     \setbox\@tempboxa\vbox to \ht\strutbox{#1\vss}%
410     \box\@tempboxa\vskip-\baselineskip
411 }

```

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

```

412 \newcommand{\mn@vadjust}[1]{%

```

```

413 \mn@zbox{\kern-\parskip
414 \leavevmode\vadjust{#1}%
415 \kern\parskip
416 }%
417 }

```

`\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`.

```

418 \providecommand*\marginfont{}{}
419 \newcommand*\raggedleftmarginnote{\raggedleft}
420 \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	
<code>\@mn@@marginnote</code> . 290	<code>\marginfont</code> 3, 418	<code>\mn@savepos</code> <u>1</u>
<code>\@mn@@marginnote</code> . . 290	<code>\marginnote</code> 2, 290	<code>\mn@strut</code> 57
<code>\@mn@atthispage</code> . . . 120	<code>\marginnoteleftadjust</code> 406	<code>\mn@vadjust</code> 412
<code>\@mn@currpage</code> 120	<code>\marginnoterightadjust</code> 406	<code>\mn@vlap</code> 408
<code>\@mn@currxpos</code> 120		<code>\mn@zbox</code> 58
<code>\@mn@ifRTL</code> 282		
<code>\@mn@marginnote</code> . . . 290	<code>\marginnotetextwidth</code> 2, 117	N
<code>\@mn@margintest</code> . . . 120	<code>\marginnotevadjust</code> 2, 406	<code>\newmarginnote</code> 116
<code>\@mn@mode@prefix</code> . . 117		
<code>\@mn@thispage</code> 120		R
	<code>\mn@abspage</code> 120	<code>\raggedleftmarginnote</code> 2, 418
I	<code>\mn@lastxpos</code> <u>1</u>	<code>\raggedrightmarginnote</code> 2, 418
<code>\if@mn@pdfmode</code> 117	<code>\mn@parboxrestore</code> . 107	
<code>\if@mn@verbose</code> 54		

Change History

v1.0a	v1.0b
<code>\marginfont</code> : Use	General: spelling fixes 1
<code>\providecommand</code> to define it. 15	v1.1
General: Example to macros	<code>\@mn@@marginnote</code> : new PDF mode feature 11
<code>\raggedleftmarginnote</code> and	<code>\@mn@currpage</code> : new (internal) . . . 7
<code>\raggedrightmarginnote</code> at	<code>\@mn@currxpos</code> : new (internal) . . . 7
documentation fixed [thanks to	<code>\@mn@margintest</code> : new PDF mode feature 7
Susumu Tanimura]. 2	

	\if@mn@pdfmode: new switch 7	v1.2	
	\marginnotetextwidth: new macro 7		\@mn@mode@prefix: (new (internal) 7
v1.1a			\if@mn@pdfmode: addition for lua _{TeX} from 0.85 7
	\if@mn@pdfmode: PDF _{TeX} since 1.40 allows \pdfsavepos in DVI mode too 7		\mn@abspage: addition for lua _{TeX} from 0.85 9
v1.1b		v1.2a	\mn@abspage: redefine \mn@currxpos only if not empty 10
	\@mn@@marginnote: use \mn@vadjust instead of \vadjust 12	v1.2b	\mn@ifRTL: new internal 11
	\if@mn@pdfmode: if level fixed 7		\mn@abspage: bidi code added . . . 10
	\mn@vadjust: new (internal) 14		General: spelling fixes (by Thomas Reuben) 1
	\mn@zbox: new (internal) 6		
v1.1c		v1.3	\@mn@mode@prefix: removed 7
	\if@mn@pdfmode: X _{TeX} has working \pdflastxpos7		\if@mn@pdfmode: removed 7
v1.1d			\marginnoteleftadjust: removed 14
	\mn@abspage: take care of \hoffset 10		\marginnoterightadjust: removed 14
	take care of \pdfhorigin 10		\mn@abspage: twocolumn test added 10
v1.1e			non PDF mode removed 9
	\@mn@@marginnote: use \mn@strut instead of \strut . 12		use new internals \mn@savepos and \mn@lastxpos 9
	\mn@strut: new (semi internal) . . . 5		\mn@lastxpos: new internal command 4
	General: new options fulladjust, heightadjust, depthadjust, and noadjust 5		\mn@savepos: new internal command 4
v1.1f			General: ε - _{TeX} removed 4
	\@mn@@marginnote: missing usage of \marginnotevadjust on left margin fixed 13		early \pdfsavepos/\savepos test 4
v1.1g		v1.4	\@mn@@marginnote: \mn@parboxrestore added . . . 13
	\@mn@@marginnote: missing \long added 11		\mn@parboxrestore: new internal command 7
	set \linewidth 13		General: new options parboxrestore (default) and noparboxrestore 7
	\@mn@@marginnote: missing \long added 11		
v1.1i		v1.4a	\mn@abspage: workaround for screwball page order 8
	\@mn@@marginnote: \ignorespaces added 13		
	\strut moved to fix hyphenation (thanks to Ulrike Fischer) 13		