

Non-Floating Margin Notes with `marginnote` Package*

Markus Kohm (inactive)[†]

September 7, 2023

Abstract

In L^AT_EX the command `\marginpar[⟨left⟩]{⟨right⟩}` can be used to create a note in the margin. But there is a problem with this command: It creates a special kind of float. So it cannot be used on floats or footnotes. The `marginnote` package supports another command `\marginnote` to create notes in the margin. This does not use any kind of float and therefore does not have the disadvantage of `\marginpar`. But there may be other problems ...

Contents

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

1 How to Use `marginnote` Package

First of all you have to load the package. You can use:

```
\usepackage{marginnote}
```

to do so. You can also use one of the following options to globally change the behaviour of `marginnote`:

`fulladjust` adjusts the margin note to the height and depth of the current line.

Note that this may sometimes add an extra height and depth to the current line, but gives the best vertical alignment. This is the default.

`heightadjust` adjusts the margin note to the height of the current line, but not the depth. Note that this can sometimes result in extra height and vertical offset of the current line.

*This file has version 1.4c, last revised 2023/09/07. `marginnote` is version 1.4b, last revised 2018/08/09.

[†]From version 1.4b the package is unmaintained. But there is a public repository with issue tracker at <https://github.com/komascript/marginnote>. So everybody can contribute and reported issues will not be lost, if a new maintainer will be found.

`depthadjust` adjusts the margin note to the depth of the current line, but not the height. Note that this can sometimes result in extra depth and vertical offset of the current line.

`noadjust` does not adjust the margin note to the height or depth of the current line. Note that this often results in vertical offsets, but rarely in vertical extra space before or after the current line.

`parboxrestore` uses a reduced `\@parboxrestore` to restore the definition of `\par` and `\` and some other commands, and sets `\parindent` to 0, `\parfillskip` to 0pt plus 1fil and `\lineskip` to `\normallineskip`, `\baselineskip` to `\normalbaselineskip` and enables `\sloppy` for every margin note. This is the default since the 1.4b release of `marginnote`. Release 1.4 and 1.4a also set `\parskip` to 0, but this caused a vertical placement problem, e.g., inside lists like `itemize`.

`noparboxrestore` does not restore any definition for the margin notes. This was the behaviour before the 1.4 release.

`\marginnote` The command `\marginnote[⟨left⟩]{⟨right⟩}[⟨voffset⟩]` can be used to set a margin note using the `marginnote` package. The first optional argument and the mandatory argument are the same as for `\marginpar` from the L^AT_EX kernel. Even `\reversemarginpar` is taken into account. The note `⟨left⟩` or `⟨right⟩` is set to the current vertical position. The second optional argument `⟨voffset⟩` can 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` The `marginnote` package needs to know the real width of the text area to find the right margin. While some environments (e.g., of the `framed` package) change `\textwidth`, `marginnote` defines its own text width macro. If you change the text area after `\begin{document}` you should add

```
\edef\marginnotetextwidth{\the\textwidth}
```

after changing the text area. You may want to do this globally using `\xdef` instead of `\edef`. Most users will never need to change `\marginnotetextwidth`.

`\marginnotevadjust` In some environments the vertical adjustment of the margin note is incorrect, e.g., one baseline too low. In this case, you can use the additional optional argument of `\marginnote` each time `\marginnote` is used, or redefine `\marginnotevadjust` at the beginning of the environment. The default definition is 0pt.

`\raggedleftmarginnote` These macros define how the margin note is 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 can change this using `\renewcommand`, e.g., use

```
\renewcommand*{\raggedleftmarginnote}{}  
\renewcommand*{\raggedrightmarginnote}{\centering}
```

to get justified text on the left and centered text on the right margin.

`\marginfont` This macro sets the font to be used for margin notes. The default is `\normalcolor`. You may can `\renewcommand` to change this, e.g., use

```
\renewcommand*{\marginfont}{\color{red}\sffamily}
```

to get red colored margin notes in the sans serif font family. Don't forget: You need to load the `color` package or the `xcolor` package to use `\color`.

2 Issues Using `marginnote`

Some known issues are listed below. For new issues, it is recommended to either discuss them in a public L^AT_EX forum or report them in `ssue` tracker on github.

- Currently, the package lacks a new maintainer. The author no longer maintains the package anymore, as it was originally intended to be used as a small hack for for a few cases. But over the years it has grown and become a conglomerate of of hacks that hack hacks that are very widely used. A break with a complete overhaul of the design would be necessary. But nothing I can or want to do at the moment. If you want to do it, please **B**ut even without taking over maintenance, you can contribute, e.g. by participating in the issue tracker of the package.
- You cannot use paragraphs with `\par` or empty lines inside a margin note. In my opinion this would note make sense. However, if you really need paragraphs you can try using `\endgraf` instead of `\par`.
- From version 1.4a, a workaround is available for double-sided documents with consecutive odd or even pages. However, it is not advisable to use double-sided documents with such page sequences as printing such documents can be problematic. The `marginnote` package displays a warning message whenever it detects those page sequences.
- From version 1.3, the `marginnote` package no longer support T_EX engines that lack primitives `\pdfsavepos` and `\pdflastxpos`, or `\savepos` and `\lastxpos`. The previous fallback option for manual adjustment has been removed. You will receive an error message if you try to use a T_EX engine without these primitives, and ϵ -T_EX primitives are also required. Nonetheless, this should not be an issue with current, free T_EX distributions like MiK_TE_X or T_EXLive.
- When using double-side layout, such as with using the `twoside` class option, `\marginnote` requires page number information to determine if a page is odd or even, and therefore wich margin to use—left or right. L^AT_EX operates using asynchronous output. Because of this, the counter `page` should not be used to obtain the current page number, unless at the header or footer of the page. To resolve the issue, the `marginnote` package employs a mechanism akin to labels. However, this implies that the accurate margin will not be determined during this L^AT_EX run, but in the next. Therefore, after making any changes to the margin notes or page breaks, two L^AT_EX runs are necessary to ensure all margins are correct.
- The `\marginnote` command utilizes `\strut` and `\vadjust` to position the margin note accurately. However, it may fail under certain circumstances. You may adjust the vertical position of the margin note using the second optional argument of `\marginnote`. Sometimes, the use of `\marginnote` command can cause the text outside to shift unnecessarily. You can choose from

the package options `fulladjust`, `heightadjust`, `depthadjust`, or `noadjust` to globally adjust, or locally redefine `\mn@strut` or `\mn@zbox`.

Note: The margin note always 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!

- From version 1.1b, `\marginnote` commands inserted between paragraphs (or in \TeX terms: *in vertical mode*) will result in the note being placed between the paragraphs instead of at the end of the preceding paragraph. You may use `\leavevmode` or the third optional argument of `\marginnote` to place it differently.
- No page break can occur within a margin note created using the `\marginnote` command.
- The use of `\marginnote` immediately after `\item` results in a different behavior compared to `\marginpar`. This is not a bug, but rather a deliberate feature!
- With math, using the `\marginnote` command may work or not work, depending on the math environment.
- If the horizontal position of the margin note is not correct, try another \LaTeX run.
- Sometimes lines may stretch vertically when using `\marginnote`, particularly if you use `\marginnote` within a list *and* upper case umlauts like “Ü”, or if you have `\lineskiplimit>0pt`. To resolve this issue, you can set `\lineskiplimit=0pt` or `\lineskiplimit=-\maxdimen` or use other available package options.
- You should not use `\marginnote` within the optional argument of `\item`.
- If `\if@twocolumn` is `\iftrue`, e.g., because you are using the `twocolumn` class option or the `\twocolumn` command, `\marginnote` decides whether the note should be placed to the left or right the columns by comparing the current horizontal position with `\columnwidth+\columnsep`. If the current horizontal position is somewhere on the left column, the note is placed in the left margin. If the current horizontal position is somewhere right of the left column and therefore on the right column, the note is placed in the right margin. However, supporting `twocolumn` mode is as problematic as supporting reverse margin notes. It is not preferred. It may be revised in the future. The current support for `twocolumn` mode has only been implemented due to a feature request from 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 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@verbostrue
55 \DeclareOption{verbose}{\@mn@verbostrue}
56 \DeclareOption{quiet}{\@mn@verbostrue}
```

`\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{%
110     \@tempskipa\parskip
111     \@parboxrestore
112     \parskip\@tempskipa
113   }%
114 }
115 \DeclareOption{noparboxrestore}{%
116   \renewcommand*\mn@parboxrestore{}%
117 }

118 \ExecuteOptions{verbose,fulladjust,parboxrestore}
119 \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.

```

120 \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.

```

121 \newcommand*\marginnotetextwidth{}
122 \let\marginnotetextwidth\textwidth
123 \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@tempswa`. To avoid changes on the last page if there is a new note on the first page, try to count the notes by page. We know that this can not be 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 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 the real x position may be written with the page label and used to calculate the correct horizontal offset.

```

124 \newcommand*\@mn@thispage{}
125 \newcommand*\@mn@currpage{}
126 \newcommand*\@mn@currxpos{}
127 \newcounter{mn@abspage}
128 \AtBeginDocument{\setcounter{mn@abspage}{1}}%
129 \g@addto@macro\@outputpage{%
130   \stepcounter{mn@abspage}%

```

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

```

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

```



```

172         }%
173         \endgroup
174         \fi
175         \PackageInfo{marginnote}{Using workaround for absolute page number}%
176         \stepcounter{mn@abspage}%
177         \fi
178     \fi
179 }%
180 }
181 \newcommand*{\@mn@margintest}{%

```

Number of the next margin note at this page.

```

182 \expandafter\ifx\csname @mn@thispage\endcsname\empty
183 \gdef\@mn@atthispage{1}%
184 \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
185 \begingroup
186 \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
187 \xdef\@mn@atthispage{\the\@tempcnta}%
188 \endgroup
189 \else
190 \gdef\@mn@atthispage{1}%
191 \fi
192 \fi
193 \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.

```

194 \let\@mn@currpage\relax
195 \let\@mn@currxpos\relax
196 \mn@savepos
197 \protected@write\@auxout{\let\themn@abspage\relax}{%
198 \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
199 {\themn@abspage}{\noexpand\number\mn@lastxpos sp}}%
200 }%

```

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.

```

201 \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.

```

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

```

```

216 \else
217 \edef\@mn@currpage{\csname
218 mn@note.\@mn@thispage.\@mn@atthispage\endcsname}%
219 \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`.

```

220 \ifx\@mn@currxpos\@empty\else
221 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\hoffset\relax}%
222 \begingroup\expandafter\expandafter\expandafter\endgroup
223 \expandafter\ifx\csname pdfhorigin\endcsname\relax\else
224 \begingroup\expandafter\expandafter\expandafter\endgroup
225 \expandafter\ifx\csname pdfoutput\endcsname\relax
226 \begingroup\expandafter\expandafter\expandafter\endgroup
227 \expandafter\ifx\csname outputmode\endcsname\relax\else
228 \ifnum \outputmode=1 %
229 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
230 +1in\relax}%
231 \fi
232 \fi
233 \else
234 \ifnum \pdfoutput=1 %
235 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
236 +1in\relax}%
237 \fi
238 \fi
239 \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).

```

240 \ifdefined\mn@pagewidth
241 \@mn@if@RTL{%
242 \PackageInfo{marginnote}{Margin note
243 \@mn@thispage.\@mn@atthispage\space in RTL mode}%
244 \edef\@mn@currxpos{%
245 \the\dimexpr\mn@pagewidth-\@mn@currxpos\relax
246 }%
247 }{%
248 \fi
249 \fi
250 \edef\@mn@currpage{\expandafter\@firstoftwo\@mn@currpage}%
251 \if@mn@verbose
252 \PackageInfo{marginnote}{Margin note \@mn@thispage.\@mn@atthispage\space
253 is on absolute page \@mn@currpage}%
254 \fi
255 \if@twoside
256 \ifodd\@mn@currpage\relax
257 \@tempwattrue
258 \if@twocolumn
259 \ifdim \@mn@currxpos

```

```

260         < \dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
261         \@tempwafalse
262         \fi
263     \fi
264 \else
265     \@tempwafalse
266     \if@twocolumn
267         \ifdim\@mn@currxpos>\dimexpr\evensidemargin+\columnwidth\relax
268             \@tempwatrue
269         \fi
270     \fi
271 \fi
272 \else
273     \if@mn@verbose
274         \PackageInfo{marginnote}{right page because not two side mode}%
275     \fi
276     \@tempwatrue
277     \if@twocolumn
278         \ifdim \@mn@currxpos
279             < \dimexpr\oddsidemargin+\columnwidth+\columnsep\relax
280             \@tempwafalse
281         \fi
282     \fi
283 \fi
284 \fi
285 }

```

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

```

286 \newcommand*{\@mn@ifRTL}{%
287 \begingroup\expandafter\expandafter\expandafter\endgroup
288 \expandafter\ifx\csname if@RTL\endcsname\iftrue
289 \expandafter\@firstoftwo
290 \else
291 \expandafter\@secondoftwo
292 \fi
293 }

```

`\marginnote` Command `\marginnote` is the main macro of the package. The others are helpers

`\@mn@marginnote` to manage the optional arguments.

```

\@mn@marginnote 294 \newcommand*{\marginnote}{%
\@mn@@marginnote 295 \@dblarg\@mn@marginnote
296 }
297 \newcommand{\@mn@marginnote}[2] []{%
298 \ifhmode
299 \@bsphack
300 \begingroup
301 \ifdim\@savsk>\z@\else
302 \def\:\{\@xifnch}\expandafter\def\: { \futurelet\@let@token\@ifnch}%
303 \fi
304 \else
305 \begingroup
306 \fi
307 \@ifnextchar [ {\@mn@marginnote[#1]#2}]{\@mn@marginnote[#1]#2[\z@]}%
308 }

```

```

309 \newcommand{\@mn@marginnote}{}
310 \long\def\@mn@marginnote[#1]#2[#3]{%
311 \endgroup

```

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

```

312 \ifhmode
313 \@mn@marginnote[#{1}]#{2}[#{3}]%
314 \@esphack
315 \else
316 \@mn@marginnote[#{1}]#{2}[#{3}]%
317 \fi
318 }
319 \newcommand{\@mn@@marginnote}{}
320 \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.

```

321 \begingroup
322 \ifhmode\mn@strut\let\@tempa\mn@vadjust\else
323 \if@inlabel\leavevmode\fi
324 \ifhmode\mn@strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
325 \fi
326 \@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.

```

327 \vbox to\z@{%
328 \vss
329 \@mn@marginnote
330 \if@reversemargin\if@tempswa
331 \@tempswafalse
332 \else
333 \@tempswatrue
334 \fi\fi
335 \if@tempswa
336 \rlap{%

```

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

```

337 \if@mn@verbose
338 \PackageInfo{marginnote}{xpos seems to be \@mn@currxpos}%
339 \fi
340 \begingroup
341 \ifx\@mn@currxpos\relax\else\ifx\@mn@currxpos\@empty\else
342 \kern-\dimexpr\@mn@currxpos\relax
343 \fi\fi
344 \ifx\@mn@currpage\relax
345 \let\@mn@currpage\@ne

```

```

346         \fi
347         \if@twoside\ifodd\@mn@currpage\relax
348             \kern\oddsidemargin
349         \else
350             \kern\evensidemargin
351         \fi
352     \else
353         \kern\oddsidemargin
354     \fi
355     \kern 1in
356 \endgroup
357 \kern\marginnotetextwidth\kern\marginparsep
358 \vbox to\z@{\kern\marginnotevadjust\kern #3
359     \vbox to\z@{
360         \hsize\marginparwidth
361         \linewidth\hsize

```

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

```

362         \kern-\parskip
363         \mn@parboxrestore
364         \marginfont\raggedrightmarginnote\strut\hspace{\z@}%
365         \ignorespaces#2\endgraf
366         \vss}%
367     \vss}%
368 }%
369 \else

```

Using the left margin.

```

370     \llap{
371         \vbox to\z@{\kern\marginnotevadjust\kern #3
372             \vbox to\z@{
373                 \hsize\marginparwidth
374                 \linewidth\hsize

```

Same like above for left margins.

```

375         \kern-\parskip
376         \mn@parboxrestore
377         \marginfont\raggedleftmarginnote\strut\hspace{\z@}%
378         \ignorespaces#1\endgraf
379         \vss
380     }%
381     \vss
382 }%
383 \if@mn@verbose
384     \PackageInfo{marginnote}{xpos seems to be \@mn@currxpos}%
385 \fi
386 \begingroup
387     \ifx\@mn@currxpos\relax\else\ifx\@mn@currpos\@empty\else
388         \kern\@mn@currxpos
389     \fi\fi
390     \ifx\@mn@currpage\relax
391         \let\@mn@currpage\@one
392     \fi
393     \if@twoside\ifodd\@mn@currpage\relax

```

```

394             \kern-\oddsidemargin
395             \else
396             \kern-\evensidemargin
397             \fi
398         \else
399             \kern-\oddsidemargin
400             \fi
401             \kern-1in
402         \endgroup
403     \kern\marginparsep
404     }%
405     \fi
406     }%
407     }%
408 \endgroup
409 }

```

`\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.

```

410 \newcommand*{\marginnotevadjust}{ }
411 \let\marginnotevadjust\z@

```

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

```

412 \newcommand{\mn@vlap}[1]{%
413   \setbox\@tempboxa\vbox to \ht\strutbox{#1\vss}%
414   \box\@tempboxa\vskip-\baselineskip
415 }

```

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

```

416 \newcommand{\mn@vadjust}[1]{%
417   \mn@zbox{\kern-\parskip
418     \leavevmode\vadjust{#1}}%
419   \kern\parskip
420 }%
421 }

```

`\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 change loading e.g. package `ragged2e`.

```

422 \providecommand*{\marginfont}{}
423 \newcommand*{\raggedleftmarginnote}{\raggedleft}
424 \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

`\@mn@@@marginnote` . [294](#)

<code>\@mn@@marginnote</code> .. 294	M	<code>\mn@savepos</code>	1
<code>\@mn@atthispage</code> ... 124	<code>\marginfont</code>	<code>\mn@strut</code>	57
<code>\@mn@currpage</code>	2, 422	<code>\mn@vadjust</code>	416
<code>\@mn@currxpos</code>	2, 294	<code>\mn@vlap</code>	412
<code>\@mn@ifRTL</code>	410	<code>\mn@zbox</code>	58
<code>\@mn@marginnote</code> ... 294	<code>\marginnoteleftadjust</code>		
<code>\@mn@margintest</code> ... 124	N	
<code>\@mn@mode@prefix</code> .. 121	410	<code>\newmarginnote</code>	120
<code>\@mn@thispage</code>	2, 121		
	<code>\marginnoterightadjust</code>	R	
I	<code>\raggedleftmarginnote</code>	
<code>\if@mn@pdfmode</code> 121	410	2, 422
<code>\if@mn@verbose</code> 54	<code>\mn@abspage</code>	<code>\raggedrightmarginnote</code>	
	124	2, 422
	<code>\mn@lastxpos</code>		
	1		
	<code>\mn@parboxrestore</code> .		
	107		

Change History

v1.0a	General: Example to macros <code>\raggedleftmarginnote</code> and <code>\raggedrightmarginnote</code> at documentation fixed [thanks to Susumu Tanimura].	2	v1.1d	<code>\mn@abspage</code> : take care of <code>\hoffset</code>	10
	<code>\marginfont</code> : Use <code>\providecommand</code> to define it.	14		take care of <code>\pdfhorigin</code>	10
v1.0b	General: spelling fixes	1	v1.1e	General: new options <code>fulladjust</code> , <code>heightadjust</code> , <code>depthadjust</code> , and <code>noadjust</code>	6
v1.1	<code>\@mn@@@marginnote</code> : new PDF mode feature	11		<code>\@mn@@@marginnote</code> : use <code>\mn@strut</code> instead of <code>\strut</code> ..	12
	<code>\@mn@currpage</code> : new (internal) ...	7		<code>\mn@strut</code> : new (semi internal) ..	6
	<code>\@mn@currxpos</code> : new (internal) ...	7	v1.1f	<code>\@mn@@@marginnote</code> : missing usage of <code>\marginnotevadjust</code> on left margin fixed	13
	<code>\@mn@margintest</code> : new PDF mode feature	7	v1.1g	<code>\@mn@@@marginnote</code> : missing <code>\long</code> added	11
	<code>\if@mn@pdfmode</code> : new switch	7		set <code>\linewidth</code>	13
	<code>\marginnotetextwidth</code> : new macro	7		<code>\@mn@@@marginnote</code> : missing <code>\long</code> added	11
v1.1a	<code>\if@mn@pdfmode</code> : PDF _T _E X since 1.40 allows <code>\pdfsavepos</code> in DVI mode too	7	v1.1i	<code>\@mn@@@marginnote</code> : <code>\ignorespaces</code> added	13
v1.1b	<code>\@mn@@@marginnote</code> : use <code>\mn@vadjust</code> instead of <code>\vadjust</code>	12		<code>\strut</code> moved to fix hyphenation (thanks to Ulrike Fischer)	13
	<code>\if@mn@pdfmode</code> : if level fixed	7	v1.2	<code>\@mn@mode@prefix</code> : (new (internal)	7
	<code>\mn@vadjust</code> : new (internal)	14		<code>\if@mn@pdfmode</code> : addition for lua _T _E X from 0.85	7
	<code>\mn@zbox</code> : new (internal)	6		<code>\mn@abspage</code> : addition for lua _T _E X from 0.85	9
v1.1c	<code>\if@mn@pdfmode</code> : X _T _E Xhas working <code>\pdflastxpos</code>	7			

v1.2a		use new internals <code>\mn@savepos</code>	
	<code>\mn@abspage</code> : redefine	and <code>\mn@lastxpos</code>	9
	<code>\@mn@currxpos</code> only if not	<code>\mn@lastxpos</code> : new internal	
	empty	command	4
v1.2b		<code>\mn@savepos</code> : new internal	
	General: spelling fixes (by Thomas	command	4
	Reuben)	v1.4	
	<code>\mn@abspage</code> : bidi code added . . .	General: new options	
	<code>\@mn@ifRTL</code> : new internal	<code>parboxrestore</code> (default) and	
v1.3		<code>noparboxrestore</code>	7
	General: ε -TeX removed	<code>\@mn@@@marginnote</code> :	
	early <code>\pdfsavepos/\savepos</code>	<code>\mn@parboxrestore</code> added . . .	13
	test	<code>\mn@parboxrestore</code> : new internal	
	<code>\@mn@mode@prefix</code> : removed	command	7
	<code>\if@mn@pdfmode</code> : removed	v1.4a	
	<code>\marginnoteleftadjust</code> : removed	<code>\mn@abspage</code> : workaround for	
	<code>\marginnoterightadjust</code> :	screwball page order	8
	removed	v1.4c	
	<code>\mn@abspage</code> : twocolumn test	General: some documentation	
	added	corrections only	1
	non PDF mode removed		