

Non-Floating Margin Notes with `marginnote` Package*

Markus Kohm (inactive)^{†‡}

June 15, 2026

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 1.5, last revised 2026-06-15. `marginnote` is version 1.5, last revised 2026-06-15.

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

[‡]Support for tagging was added by Ulrike Fischer in version 1.4e on June 13, 2026.

`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 [the project's](#) issue tracker on Codeberg.

- 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 a few cases. But over the years it has grown and become a conglomerate 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 [contact the author](#). But 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 not 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 supports T_EX engines that lack primitives `\pdfsavepos` and `\pdfsavepos`, 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_T_EX 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 which 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 `left margin+\columnwidth+.5\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@verbosetrue
55 \DeclareOption{verbose}{\@mn@verbosetrue}
56 \DeclareOption{quiet}{\@mn@verbosetrue}
```

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

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

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

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

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

```
59 \DeclareOption{fulladjust}{%
60   \renewcommand*\mn@strut{\strut}%
61   \renewcommand{\mn@zbox}[1]{}%
62   \bgroup
```

Avoid the execution of the `para/end` hooks here (e.g. wrong for tagging):

```
63   \setbox\@tempboxa\vbox{#1\RawParEnd}%
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
```

Avoid the execution of the `para/end` hooks here, this is wrong for tagging:

```
75   \setbox\@tempboxa\vbox{#1\RawParEnd}%
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
```

Avoid the execution of the `para/end` hooks here, this is wrong for tagging:

```
87   \setbox\@tempboxa\vbox{#1\RawParEnd}%
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

```

Avoid the execution of the para/end hooks here, this is wrong for tagging:

```

99      \setbox\@tempboxa\vbox{\kern-\ht\strutbox #1\RawParEnd}%
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 don't 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}{%

```

Add tagging support (perhaps not really needed but safer):

```

110     \UseTaggingSocket{para/restore}%
111     \@tempskipa\parskip
112     \@parboxrestore
113     \parskip\@tempskipa
114   }%
115 }

```

```

116 \DeclareOption{noparboxrestore}{%
117   \renewcommand*{\mn@parboxrestore}{%

```

Tagging should better restore:

```

118     {\UseTaggingSocket{para/restore}}%
119 }

```

```

120 \ExecuteOptions{verbose,fulladjust,parboxrestore}
121 \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.

```

122 \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.

```

123 \newcommand*{\marginnotetextwidth}{%
124 \let\marginnotetextwidth\textwidth
125 \AtBeginDocument{\edef\marginnotetextwidth{\the\textwidth}}

\@mn@margintest Macro \@mn@margintest does the complete test, which margin to use. The result
\@mn@thispage may be found at \if@tempswa. To avoid changes on the last page if there is a
\@mn@atthispage new note on the first page, try to count the notes by page. We know that this
\@mn@currpage can not be successful, but never the less it may be a good try. \@mn@thispage
\@mn@currxpos saves the page number of the last usage of \@mn@margintest. \@mn@atthispage
\mn@abspage 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.

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

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

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

```



```

163         \advance\c@page\m@ne
164         \PackageWarningNoLine{marginnote}{%
165             Consecutive even pages found.\MessageBreak
166             Note, it is not recommended to use consecutive\MessageBreak
167             even pages in a double-ended document.\MessageBreak
168             The pages of your document should always\MessageBreak
169             be a sequence: odd-even-odd-even-...\MessageBreak
170             Maybe you've forgotten a
171             \@ifundefined{KOMAClassName}{%
172                 {\string\cleardoublepage}%
173                 {\string\cleardoubleevenpage}%
174             before\MessageBreak
175             changing the page numbering on page \thepage
176             }%
177         \endgroup
178     \fi
179     \if@mn@verbose
180         \PackageInfo{marginnote}{Using workaround for absolute page number}%
181     \fi
182     \stepcounter{mn@abspage}%
183 \fi
184 \fi
185 }%
186 }
187 \newcommand*{\@mn@margintest}{%

```

Number of the next margin note at this page.

```

188 \expandafter\ifx\csname @mn@thispage\endcsname\empty
189 \gdef\@mn@atthispage{1}%
190 \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
191     \begingroup
192     \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
193     \xdef\@mn@atthispage{\the\@tempcnta}%
194     \endgroup
195 \else
196     \gdef\@mn@atthispage{1}%
197 \fi
198 \fi
199 \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.

```

200 \let\@mn@currpage\relax
201 \let\@mn@currxpos\relax
202 \mn@savepos
203 \protected@write\@auxout{\let\themn@abspage\relax}{%
204     \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
205         {\themn@abspage}{\noexpand\number\mn@lastxpos sp}}}%
206 }%

```

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.

```

207 \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.

```

208 \if@twoside
209 \if@mn@verbose
210 \PackageInfo{marginnote}{Suggest that margin
211 note \@mn@thispage.\@mn@atthispage\space will be on\MessageBreak
212 absolute page \themn@abspage.\MessageBreak
213 This may be wrong}%
214 \fi
215 \ifodd\value{mn@abspage}\@tempwattrue\else\@tempwafalse\fi
216 \else
217 \if@mn@verbose
218 \PackageInfo{marginnote}{right page because not two side mode}%
219 \fi
220 \@tempwattrue
221 \fi
222 \else
223 \edef\@mn@currpage{\csname
224 mn@note.\@mn@thispage.\@mn@atthispage\endcsname}%
225 \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`.

```

226 \ifx\@mn@currxpos\empty\else
227 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\hoffset\relax}%
228 \begingroup\expandafter\expandafter\expandafter\endgroup
229 \expandafter\ifx\csname pdfhorigin\endcsname\relax\else
230 \begingroup\expandafter\expandafter\expandafter\endgroup
231 \expandafter\ifx\csname pdfoutput\endcsname\relax
232 \begingroup\expandafter\expandafter\expandafter\endgroup
233 \expandafter\ifx\csname outputmode\endcsname\relax\else
234 \ifnum \outputmode=1 %
235 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
236 +1in\relax}%
237 \fi
238 \fi
239 \else
240 \ifnum \pdfoutput=1 %
241 \edef\@mn@currxpos{\the\dimexpr \@mn@currxpos -\pdfhorigin
242 +1in\relax}%
243 \fi
244 \fi
245 \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).

```

246 \ifdefined\mn@pagewidth
247 \@mn@if@RTL{%
248 \if@mn@verbose
249 \PackageInfo{marginnote}{Margin note
250 \@mn@thispage.\@mn@atthispage\space in RTL mode}%
251 \fi

```

```

252         \edef\@mn@curr xpos{%
253             \the\dim expr\mn@pagewidth-\@mn@curr xpos\relax
254         }%
255     }{}%
256 \fi
257 \fi
258 \edef\@mn@currpage{\expandafter\@firstoftwo\@mn@currpage}%
259 \if@mn@verbose
260     \PackageInfo{marginnote}{Margin note \@mn@thispage.\@mn@atthispage\space
261         is on absolute page \@mn@currpage}%
262 \fi
263 \if@twoside
264     \ifodd\@mn@currpage\relax
265         \@temp swat true
266         \if@twocolumn
267             \ifdim \@mn@curr xpos
268                 < \dim expr 1in+\oddsidemargin+\columnwidth+.5\columnsep\relax
269                 \@temp swa false
270             \fi
271         \fi
272     \else
273         \@temp swa false
274         \if@twocolumn
275             \ifdim \@mn@curr xpos>\dim expr 1in+\evensidemargin+.5\columnwidth\relax
276                 \@temp swat true
277             \fi
278         \fi
279     \fi
280 \else
281     \if@mn@verbose
282         \PackageInfo{marginnote}{right page because not two side mode}%
283     \fi
284     \@temp swat true
285     \if@twocolumn
286         \ifdim \@mn@curr xpos
287             < \dim expr 1in+\oddsidemargin+\columnwidth+.5\columnsep\relax
288             \@temp swa false
289         \fi
290     \fi
291 \fi
292 \fi
293 }

```

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

```

294 \newcommand*{\@mn@if@RTL}{%
295     \begingroup\expandafter\expandafter\expandafter\endgroup
296     \expandafter\ifx\csname if@RTL\endcsname\iftrue
297         \expandafter\@firstoftwo
298     \else
299         \expandafter\@secondoftwo
300     \fi
301 }

```

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

```

\@mn@marginnote
\@mn@@marginnote
\@mn@@@marginnote

```

to manage the optional arguments.

```

302 \newcommand*{\marginnote}{%
303   \@dblarg\@mn@marginnote
304 }
305 \newcommand{\@mn@marginnote}[2][{}]{%
306   \ifhmode
307     \@bsphack
308     \begingroup
309     \ifdim\@savsk>\z@ \else
310       \def\:\{\@xifnch}\expandafter\def\:\: { \futurelet\@let@token\@ifnch}%
311       \fi
312     \else
313       \begingroup
314       \fi
315     \@ifnextchar [{\@mn@marginnote[{#1}]{#2}}{\@mn@marginnote[{#1}]{#2}[\z@]}%
316 }
317 \newcommand{\@mn@@marginnote}{%
318 \long\def\@mn@@marginnote[#1]#2[#3]{%
319   \endgroup

```

Tagging support:

```

320 \UseTaggingSocket{marginpar/begin}%

```

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

```

321 \ifhmode
322   \@mn@@marginnote[{#1}]{#2}[{#3}]%
323   \@esphack
324 \else
325   \@mn@@marginnote[{#1}]{#2}[{#3}]%
326 \fi

```

Tagging support:

```

327 \UseTaggingSocket{marginpar/end}%
328 }
329 \newcommand{\@mn@@marginnote}{%
330 \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 to 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.

```

331 \begingroup
332   \ifmmode\mn@strut\let\@tempa\mn@vadjust\else
333     \if@inlabel\leavevmode\fi
334     \ifhmode\mn@strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
335   \fi
336   \@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 places to the margin.

```

337      \vbox to\z@{%
338      \vss
339      \@mn@margintest
340      \if@reversemargin\if@tempswa
341      \@tempswafalse
342      \else
343      \@tempswatrue
344      \fi\fi
345      \if@tempswa
346      \rlap{%

```

If \@mn@currxpos is neither \relax nor empty it is the real current x position of the last PDF_{La}T_EX run and may be used to calculate the real horizontal offset.

```

347      \if@mn@verbose
348      \PackageInfo{marginnote}{xpos seems to be \@mn@currxpos}%
349      \fi
350      \begingroup
351      \ifx\@mn@currxpos\relax\else\ifx\@mn@currxpos\@empty\else
352      \kern-\dimexpr\@mn@currxpos\relax
353      \fi\fi
354      \ifx\@mn@currpage\relax
355      \let\@mn@currpage\@one
356      \fi
357      \if@twoside\ifodd\@mn@currpage\relax
358      \kern\oddsidemargin
359      \else
360      \kern\evensidemargin
361      \fi
362      \else
363      \kern\oddsidemargin
364      \fi
365      \kern 1in
366      \endgroup
367      \kern\marginnotetextwidth\kern\marginparsep
368      \vbox to\z@{\kern\marginnotevadjust\kern #3
369      \vbox to\z@{%
370      \hsize\marginparwidth
371      \linewidth\hsize

```

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

```

372      \kern-\parskip
373      \mn@parboxrestore
374      \marginfont\raggedrightmarginnote\strut\hspace{\z@}%
375      \ignorespaces#2\endgraf
376      \vss}%
377      \vss}%
378      }%
379      \else

```

Using the left margin.

```

380      \llap{%
381      \vbox to\z@{\kern\marginnotevadjust\kern #3
382      \vbox to\z@{%
383      \hsize\marginparwidth

```

```

384          \linewidth\hsize
Same like above for left margins.
385          \kern-\parskip
386          \mn@parboxrestore
387          \marginfont\raggedleftmarginnote\strut\hspace{\z@}%
388          \ignorespaces#1\endgraf
389          \vss
390      }%
391      \vss
392  }%
393  \if@mn@verbose
394      \PackageInfo{marginnote}{xpos seems to be \mn@currxpos}%
395  \fi
396  \begingroup
397      \ifx\mn@currxpos\relax\else\ifx\mn@currpos\@empty\else
398          \kern\mn@currxpos
399      \fi\fi
400      \ifx\mn@currpage\relax
401          \let\mn@currpage\@ne
402      \fi
403      \if@twoside\ifodd\mn@currpage\relax
404          \kern-\oddsidemargin
405      \else
406          \kern-\evensidemargin
407      \fi
408      \else
409          \kern-\oddsidemargin
410      \fi
411      \kern-1in
412  \endgroup
413  \kern\marginparsep
414  }%
415  \fi
416  }%
417  }%
418  \endgroup
419 }

```

`\marginnoterightadjust`

`\marginnoteleftadjust`

`\marginnotevadjust` This may be used to define an automatic vertical adjust. The default is zero. Values greater than zero will move the margin note down, values less than zero will move the margin note up.

```
420 \newcommand*{\marginnotevadjust}{}
421 \let\marginnotevadjust\z@
```

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

```

422 \newcommand{\mn@vlap}[1]{%
423   \setbox\@tempboxa\vbox to \ht\strutbox{#1\vss}%
424   \box\@tempboxa\vskip-\baselineskip
425 }

```

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

```

426 \newcommand{\mn@vadjust}[1]{%
427   \mn@zbox{\kern-\parskip

```

Avoid the execution of the para hooks here (wrong e.g. with tagging):

```

428   \RawIndent
429   \vadjust{#1}%
430   \kern\parskip
431 }%
432 }

```

`\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`.

```

433 \providecommand*{\marginfont}{}
434 \newcommand*{\raggedleftmarginnote}{\raggedleft}
435 \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> . 302	<code>\marginfont</code> 2 , 433	<code>\mn@savepos</code> 1
<code>\@mn@marginnote</code> . . 302	<code>\marginnote</code> 2 , 302	<code>\mn@strut</code> 57
<code>\@mn@atthispage</code> . . . 126	<code>\marginnoteleftadjust</code> 420	<code>\mn@vadjust</code> 426
<code>\@mn@currpage</code> 126	<code>\marginnoterightadjust</code> 420	<code>\mn@vlap</code> 422
<code>\@mn@currxpos</code> 126	<code>\marginnotetextwidth</code> 2 , 123	<code>\mn@zbox</code> 58
<code>\@mn@ifRTL</code> 294	<code>\marginnotevadjust</code> 2 , 420	
<code>\@mn@marginnote</code> . . . 302		N
<code>\@mn@margintest</code> . . . 126	<code>\mn@abspage</code> 126	<code>\newmarginnote</code> 122
<code>\@mn@mode@prefix</code> . . 123	<code>\mn@lastxpos</code> 1	
<code>\@mn@thispage</code> 126	<code>\mn@parboxrestore</code> . 107	R
		<code>\raggedleftmarginnote</code> 2 , 433
I		<code>\raggedrightmarginnote</code> 2 , 433
<code>\if@mn@pdfmode</code> 123		
<code>\if@mn@verbose</code> 54		

Change History

v1.0a	v1.0b
General: Example to macros	General: spelling fixes 1
<code>\raggedleftmarginnote</code> and	v1.1
<code>\raggedrightmarginnote</code> at	<code>\@mn@@marginnote</code> : new PDF
documentation fixed [thanks to	mode feature 11
Susumu Tanimura]. 2	<code>\@mn@currpage</code> : new (internal) . . . 8
<code>\marginfont</code> : Use	<code>\@mn@currxpos</code> : new (internal) . . . 8
<code>\providecommand</code> to define it. 15	<code>\@mn@margintest</code> : new PDF mode

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