

Non-Floating Margin Notes with `marginnote` Package*

Markus Kohm[†]

2009/02/26

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 ...

1 How to Use `marginnote` Package

First of all you have to load. You may use:

```
\usepackage{marginnote}
```

to do so.

`\marginnote` The command `\marginnote[⟨left⟩]{⟨right⟩}[⟨voffset⟩]` may be used to set a margin note using `marginnote`. The first optional argument and the mandatory argument are same using `\marginpar` from the L^AT_EX kernel. Even `\reversemarginpar` will be considered. The note `⟨left⟩` or `⟨right⟩` will be put at the current vertical position. Second optional argument `⟨voffset⟩` may be used to adjust the vertical position of the margin note. Use a negative dimension to move it up or a positive dimension to move it down.

`\marginnoteleftadjust`
`\marginnoterightadjust` At some environments, e.g. `framed` from the `framed` package the horizontal placement of the margin notes are not correct. In this case you may redefine `\marginnoteleftadjust` and `\marginnoterightadjust` to fix this. Note that these are macros not lengths! So you have to use `\renewcommand`, `\def` or `\let` to change them. You may e.g. use

```
\begingroup
```

*This file has version number v1.1c, last revised 2009/02/26.

[†]Email: komascript@gmx.info

```

\makeatletter
\g@addto@macro\framed{%
  \let\marginnoteleftadjust\FrameSep
  \let\marginnoterightadjust\FrameSep
}
\endgroup

```

at your preamble after loading package `framed` to fix the problem using `framed` environment.

NOTE: `\marginnoteleftadjust` and `\marginnoterightadjust` will be used only, if the correct horizontal position cannot be determined using PDF_TE_X features (`\pdfsavepos` and `\pdfsavepos`). So if you are using PDF_LA_TE_X with PDF output or PDF_LA_TE_X with PDF_TE_X-version since 1.40 or X_YT_EX you will not need to use the example code above, but you will need at least two L_AT_EX runs to get the correct horizontal positions of the margin notes.

`\marginnotetextwidth`

Package *marginnote* needs to know the real width of the type area to find the right margin. While some environments (e.g. those of package *framed*) change `\textwidth`, *marginnote* defines 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`
`\raggedrightmarginnote`

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

- align margin notes at the left margin right to the margin,
- align margin notes at the right margin left to the margin.

You may change this using `\renewcommand`, e.g. use

```

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

```

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

`\marginfont`

This macro defines the font that will be used to set margin notes. The default is `\normalcolor`. You may use `\renewcommand` to change this, e.g. use

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

to get red colored margin notes in sans serif font family. You need to load e.g. package `color` to use `\color`.

2 Known Problems Using `marginnote`

At double side layout (e.g. using class option `twoside`) `\marginnote` needs to know the number of the current page to decide whether the page is odd or even and so whether to use left or right margin. \LaTeX uses an asynchronous output. Because of this counter `page` should not be used to get the number of the current page unless you are at page head or foot. To solve the problem `marginnote` uses a mechanism similar to labels. But this means, that the correct margin won't be known at this \LaTeX run but only at the next. So after adding or deleting a margin note or after each change of page break you need two \LaTeX runs to get all margins right.

The command `\marginnote` uses `\strut` and `\vadjust` to put the margin note at the correct position. But under some circumstances this may fail. You may adjust the vertical position of the margin note using the second optional argument of `\marginnote`. Sometimes even the text outside `\marginnote` will be moved because of using `\marginnote`. There's currently no other solution for this problem than moving the `\marginnote` command.

Note: The margin note will be placed at the current vertical line. This means, if you are using two `\marginnote` commands at the same line, they will be put on the same place. This is not a bug but a feature!

Since release 1.1b `\marginnote` between paragraphs (in vertical mode) will place the note between the paragraphs instead of the end of the previous paragraph. You may use `\leavevmode` or the third optional argument of `\marginnote` to place it different.

No page break may occur inside a margin note created with `\marginnote`.

`\marginnote` somewhat different from `\marginpar` if used immediately after `\item`. This is not a bug, it's a feature!

With math `\marginnote` may work or may not depending on the math environment.

If you are using \XeTeX , \PDF\LaTeX since version 1.40 or \PDF\LaTeX before version 1.40 with PDF output and the horizontal position of a margin note is wrong, do one more \PDF\LaTeX run.

Sometimes lines are stretched vertically using `\marginnote`, e.g. if you're using `\marginnote` at a list *and* upper case umlauts like "Ü". In this case `\lineskiplimit=-\maxdimen` should help.

You should not use `\marginnote` at the optional argument of `\item`.

3 Implementation

First declare and process the options.

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

```
1 \newif\if@mn@verbose\@mn@verbosetrue
2 \DeclareOption{verbose}{\@mn@verbosetrue}
3 \DeclareOption{quiet}{\@mn@verbosefalse}
```

```

4 \ExecuteOptions{verbose}
5 \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.
6 \newcommand*{\newmarginnote}{\@newl@bel{mn}}

\if@mn@pdfmode We need to know, whether or not PDFTEX and which version of PDFTEX is used.
With PDFTEX the horizontal output position may be detected using \pdfsavepos
and \pdflastxpos. So the relative position of the margin may be calculated.
Without PDFTEX only manual adjustment is available. While PDF mode or not
may change before start of the document, setting up the switch is delayed.
7 \newif\if@mn@pdfmode\@mn@pdfmodefalse
8 \AtBeginDocument{%
9   \begingroup\expandafter\expandafter\expandafter\endgroup
10  \expandafter\ifx\csname pdflastxpos\endcsname\relax\else % bg or 1
11    \begingroup\expandafter\expandafter\expandafter\endgroup
12    \expandafter\ifx\csname pdftexversion\endcsname\relax % bg 2
13      \begingroup\expandafter\expandafter\expandafter\endgroup
14      \expandafter\ifx\csname pdfoutput\endcsname\relax % bg 3
15        \begingroup\expandafter\expandafter\expandafter\endgroup
16        \expandafter\ifx\csname XeTeXrevision\endcsname\relax\else % bg 4
17          \@mn@pdfmodetrue
18          \fi % ed 4
19        \else % or 3
20          \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi % bg ed 4
21        \fi % ed 3
22      \else % or 2
23        \ifnum \pdftexversion<140 % bg 3
24          \begingroup\expandafter\expandafter\expandafter\endgroup
25          \expandafter\ifx\csname pdfoutput\endcsname\relax % bg 4
26            \else % or 4
27              \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi % bg ed 5
28            \fi % ed 4
29          \else % or 3
30            \@mn@pdfmodetrue
31            \fi % ed 3
32          \fi % ed 2
33        \fi % ed 1
34      \if@mn@verbose
35        \if@mn@pdfmode
36          \PackageInfo{marginnote}{%
37            \string\pdfoutput\space not 0 and \string\pdflastxpos\space
38            available.\MessageBreak
39            Extended position detection mode activated\@gobble
40          }%
41        \else

```

```

42     \PackageInfo{marginnote}{%
43         either \string\pdflastxpos\space or \string\pdfoutput\space not
44         available\MessageBreak
45         or \string\pdfoutput\space set to 0.\MessageBreak
46         Extended position detection mode deactivated\@gobble
47     }%
48     \fi
49     \fi
50 }

```

`\marginnotetextwidth` Some environments change `\textwidth`. But at PDF mode we need to know the real text width to find the right margin. So we use our own text width macro. Sometimes it may be usefull if the user can set it up. Because of this it is a user command.

```

51 \newcommand*{\marginnotetextwidth}{}
52 \let\marginnotetextwidth\textwidth
53 \AtBeginDocument{\if@mn@pdfmode\edef\marginnotetextwidth{\the\textwidth}\fi}

```

`\@mn@margintest` Macro `\@mn@margintest` does the complete test, which margin to use. The result may be found at `\if@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 successfull, but never the less it may be a good try. `\@mn@thispage` saves the page number of the last usage of `\@mn@margintest`. `\@mn@atthispage` saves the number of margin note at this page. But we need to know the absolut page number to do this. So we increase the absolut page number `mn@abspage` at every `\@outputpage`. `\@mn@currpage` is the page from the page label if found. `\@mn@currxpos` is somehow special. Using PDF_T_EX the real *x* position may be written with the page label and used to calculate the correct horizontal offset. In this case `\marginnoteleftadjust` and `\marginnoterightadjust` will not be used!

```

54 \newcommand*{\@mn@thispage}{}
55 \newcommand*{\@mn@currpage}{}
56 \newcommand*{\@mn@currxpos}{}
57 \newcounter{mn@abspage}
58 \AtBeginDocument{\setcounter{mn@abspage}{1}%
59     \g@addto@macro\@outputpage{\stepcounter{mn@abspage}}}
60 \newcommand*{\@mn@margintest}{%

```

Number of the next margin note at this page.

```

61     \expandafter\ifx\csname @mn@thispage\endcsname\@empty
62         \gdef\@mn@atthispage{1}%
63     \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
64         \begingroup
65             \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
66             \xdef\@mn@atthispage{\the\@tempcnta}%
67         \endgroup
68     \else
69         \gdef\@mn@atthispage{1}%
70     \fi

```

```

71 \fi
72 \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.

```

73 \let\@mn@currpage\relax
74 \let\@mn@currxpos\relax
75 \if@mn@pdfmode
76   \pdfsavepos
77   \protected@write\@auxout{\let\themn@abspage\relax}{%
78     \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
79       {\themn@abspage}{\noexpand\number\pdfsavepos sp}}%
80   }%
81 \else
82   \protected@write\@auxout{\let\themn@abspage\relax}{%
83     \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
84       {\themn@abspage}{}}%
85   }%
86 \fi

```

If the margin note label was not defined, it seems to be new. In this case the absolute page number will be used for the test instead of the saved real page number.

```

87 \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.

```

88 \if@twoside
89   \if@mn@verbose
90     \PackageInfo{marginnote}{Suggest that margin
91       note \@mn@thispage.\@mn@atthispage\space will be on\MessageBreak
92       absolute page \themn@abspage.\MessageBreak
93       This may be wrong}%
94   \fi
95   \ifodd\value{mn@abspage}\@tempwattrue\else\@tempwafalse\fi
96 \else
97   \if@mn@verbose
98     \PackageInfo{marginnote}{right page because not two side mode}%
99   \fi
100   \@tempwattrue
101 \fi
102 \else
103   \edef\@mn@currpage{\csname
104     mn@note.\@mn@thispage.\@mn@atthispage\endcsname}%
105   \edef\@mn@currxpos{\expandafter\@secondoftwo\@mn@currpage}%
106   \edef\@mn@currpage{\expandafter\@firstoftwo\@mn@currpage}%
107   \if@mn@verbose
108     \PackageInfo{marginnote}{Margin note \@mn@thispage.\@mn@atthispage\space
109       is on absolute page \@mn@currpage\MessageBreak}%
110   \fi
111 \if@twoside

```

```

112     \ifodd\@mn@currpage\relax
113     \@tempwattrue
114     \else
115     \@tempwafalse
116     \fi
117   \else
118     \if@mn@verbose
119     \PackageInfo{marginnote}{right page because not two side mode}%
120     \fi
121     \@tempwattrue
122   \fi
123 \fi
124 }

```

```

\marginnote Command \marginnote is the main macro of the package. The others are helpers
\@mn@marginnote to manage the optional arguments.
\@mn@@marginnote 125 \newcommand*{\marginnote}{%
\@mn@@@marginnote 126 \@dblarg\@mn@marginnote
127 }
128 \newcommand{\@mn@marginnote}[2][{}]{%
129   \ifhmode
130     \@bsphack
131     \begingroup
132     \ifdim\@savsk>\z@\else
133       \def\:\{\@xifnch}\expandafter\def\:{ \futurelet\@let@token\@ifnch}%
134     \fi
135   \else
136     \begingroup
137   \fi
138   \@ifnextchar [{\@mn@@marginnote[#1][#2]}{\@mn@@marginnote[#1][#2][\z@]}%
139 }
140 \newcommand{\@mn@@marginnote}{}
141 \def\@mn@@marginnote[#1]#2[#3]{%
142   \endgroup

```

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

```

143   \ifhmode
144     \@mn@@@marginnote[#1][#2][#3]%
145     \@esphack
146   \else
147     \@mn@@@marginnote[#1][#2][#3]%
148   \fi
149 }
150 \newcommand{\@mn@@@marginnote}{}
151 \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.

```

152 \begingroup
153   \ifmmode\strut\let\@tempa\mn\vadjust\else
154     \if@inlabel\leavevmode\fi
155     \ifhmode\strut\let\@tempa\mn\vadjust\else\let\@tempa\mn\vlap\fi
156   \fi
157   \@tempa{%

```

Everything will be put upwards using a vbox with zero height and depth and `\vss`. At this box the margin test will be done. If `csreversemargin` was used, the logic switches. Then the note will be placed to the margin.

```

158     \vbox to\z@{%
159       \vss
160       \@mn@margintest
161       \if@reversemargin\if@tempswa
162         \@tempswafalse
163       \else
164         \@tempswatrue
165       \fi\fi
166       \if@tempswa
167         \rlap{%

```

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

```

168         \ifx\@mn@currpos\relax
169           \kern\marginnoterightadjust
170         \if@mn@verbose
171           \PackageInfo{marginnote}{%
172             xpos not known,\MessageBreak
173             using \string\marginnoterightadjust}%
174         \fi
175       \else\ifx\@mn@currpos\empty
176         \kern\marginnoterightadjust
177         \if@mn@verbose
178           \PackageInfo{marginnote}{%
179             xpos not known,\MessageBreak
180             using \string\marginnoterightadjust}%
181         \fi
182       \else
183         \if@mn@verbose
184           \PackageInfo{marginnote}{%
185             xpos seems to be \@mn@currpos,\MessageBreak
186             \string\marginnoterightadjust
187             \space ignored}%
188         \fi
189       \begingroup
190         \setlength{\@tempdima}{\@mn@currpos}%

```

```

191          \kern-\@tempdima
192          \if@twoside\ifodd\@mn@currpage\relax
193            \kern\oddsidemargin
194          \else
195            \kern\evensidemargin
196          \fi
197        \else
198          \kern\oddsidemargin
199        \fi
200        \kern 1in
201      \endgroup
202    \fi
203  \fi
204  \kern\marginnotetextwidth\kern\marginparsep
205  \vbox to\z@{\kern\marginnotevadjust\kern #3
206    \vbox to\z@{%
207      \hsize\marginparwidth

```

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

```

208          \kern-\parskip
209          \marginfont\raggedrightmarginnote\hspace{\z@}\strut#2\endgraf
210          \vss}%
211        \vss}%
212      }%
213    \else
214      \llap{%
215        \vbox to\z@{\kern #3
216          \vbox to\z@{%
217            \hsize\marginparwidth

```

Same like above for left margins.

```

218          \kern-\parskip
219          \marginfont\raggedleftmarginnote\hspace{\z@}\strut#1\endgraf
220          \vss}%
221        \vss}%
222      \ifx\@mn@currxpos\relax
223        \kern\marginnoteleftadjust
224      \if@mn@verbose
225        \PackageInfo{marginnote}{%
226          xpos not known,\MessageBreak
227          using \string\marginnoteleftadjust}%
228      \fi
229    \else\ifx\@mn@currxpos\@empty
230      \kern\marginnoteleftadjust
231    \if@mn@verbose
232      \PackageInfo{marginnote}{%
233        xpos not known,\MessageBreak
234        using \string\marginnoteleftadjust}%
235    \fi
236  \else
237    \if@mn@verbose

```

```

238         \PackageInfo{marginnote}{%
239             xpos seems to be \@mn@currxpos,\MessageBreak
240             \string\marginnoteleftadjust
241             \space ignored}%
242     \fi
243     \begingroup
244         \kern\@mn@currxpos
245         \if@twoside\ifodd\@mn@currrpage\relax
246             \kern-\oddsidemargin
247         \else
248             \kern-\evensidemargin
249         \fi
250     \else
251         \kern-\oddsidemargin
252     \fi
253     \kern-1in
254 \endgroup
255 \fi
256 \fi
257 \kern\marginparsep
258 }%
259 \fi
260 }%
261 }%
262 \endgroup
263 }

```

`\marginnoterightadjust` These may be used to define an automatic horizontal adjust. The default is zero.
`\marginnoteleftadjust` They will be used only if not PDF_TE_X or PDF_TE_X before version 1.40 in DVI mode is used, because in this case the save position features are not available.

```

264 \newcommand*{\marginnoterightadjust}{\z@}
265 \newcommand*{\marginnoteleftadjust}{\z@}
266 \let\marginnoterightadjust\z@
267 \let\marginnoteleftadjust\z@

```

`\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.

```

268 \newcommand*{\marginnotevadjust}{\z@}
269 \let\marginnotevadjust\z@

```

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

```

270 \newcommand{\mn@vlap}[1]{%
271     \setbox\@tempboxa\vbox to \ht\strutbox{#1\vss}%
272     \box\@tempboxa\vskip-\baselineskip
273 }

```

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

```

274 \newcommand{\mn@zbox}[1]{%

```

```

275 \bgroup
276 \setbox\@tempboxa\vbox{#1}%
277 \ht\@tempboxa\ht\strutbox
278 \dp\@tempboxa\dp\strutbox
279 \wd\@tempboxa\z@
280 \box\@tempboxa
281 \egroup
282 }

```

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

```

283 \newcommand{\mn@vadjust}[1]{%
284 \mn@zbox{\kern-\parskip
285 \leavevmode\vadjust{#1}%
286 \kern\parskip
287 }%
288 }

```

`\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`.

```

289 \providecommand*{\marginfont}{}
290 \newcommand*{\raggedleftmarginnote}{\raggedleft}
291 \newcommand*{\raggedrightmarginnote}{\raggedright}

```

Change History

v1.0a	General: Example to macros <code>\raggedleftmarginnote</code> and <code>\raggedrightmarginnote</code> at documentation fixed [thanks to Susumu Tanimura]. 2	<code>\if@mn@pdfmode</code> : new switch 4 <code>\marginnotetextwidth</code> : new macro 5
	<code>\marginfont</code> : Use <code>\providecommand</code> to define it. 11	v1.1a <code>\if@mn@pdfmode</code> : PDF _T _E X since 1.40 allows <code>\pdfsavepos</code> in DVI mode too 4
v1.0b	General: spelling fixes 1	v1.1b <code>\@mn@@marginnote</code> : use <code>\mn@vadjust</code> instead of <code>\vadjust</code> 7
v1.1	<code>\@mn@@marginnote</code> : new PDF mode feature 7 <code>\@mn@currpage</code> : new (internal) . . . 5 <code>\@mn@currxpos</code> : new (internal) . . . 5 <code>\@mn@margintest</code> : new PDF mode feature 5	<code>\if@mn@pdfmode</code> : if level fixed 4 <code>\mn@vadjust</code> : new (internal) 11 <code>\mn@zbox</code> : new (internal) 10 v1.1c <code>\if@mn@pdfmode</code> : X _q T _E X has work- ing <code>\pdflastxpos</code> but no <code>\pdfTeXversion</code> 4