

Non-Floating Margin Notes with `marginnote` Package*

Markus Kohm[†]

2009/02/16

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

```
\begin{group}
```

*This file has version number v1.1b, last revised 2009/02/16.

[†]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 you will not need to use the example code above, but you will need at least two PDF_LA_TE_X 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 \LaTeX since version 1.40 or \LaTeX before version 1.40 with PDF output and the horizontal position of a margin note is wrong, do one more \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@verbosetrue}
```

```

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\else% bg or 3
15        \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi% bg ed 4
16        \fi% ed 3
17      \else% or 2
18        \ifnum \pdftexversion<140% bg 3
19          \begingroup\expandafter\expandafter\expandafter\endgroup
20          \expandafter\ifx\csname pdfoutput\endcsname\relax\else% bg or 4
21            \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi% bg ed 5
22            \fi% ed 4
23          \else% or 3
24            \@mn@pdfmodetrue
25            \fi% ed 3
26          \fi% ed 2
27        \fi% ed 1
28      \if@mn@verbose
29        \if@mn@pdfmode
30          \PackageInfo{marginnote}{%
31            \string\pdfoutput\space not 0 and \string\pdflastxpos\space
32            available.\MessageBreak
33            Extended position detection mode activated\@gobble
34          }%
35        \else
36          \PackageInfo{marginnote}{%
37            either \string\pdflastxpos\space or \string\pdfoutput\space not
38            available.\MessageBreak
39            or \string\pdfoutput\space set to 0.\MessageBreak
40            Extended position detection mode deactivated\@gobble
41          }%

```

```

42   \fi
43   \fi
44 }

```

`\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.

```

45 \newcommand*{\marginnotetextwidth}{%
46 \let\marginnotetextwidth\textwidth
47 \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_E_X the real x position may be written with the page label and used to calculate the correct horizontal offset. In this case `\marginnoteleftadjust` and `\marginnoterightadjust` will not be used!

```

48 \newcommand*{\@mn@thispage}{%
49 \newcommand*{\@mn@currpage}{%
50 \newcommand*{\@mn@currxpos}{%
51 \newcounter{mn@abspage}
52 \AtBeginDocument{\setcounter{mn@abspage}{1}}%
53 \g@addto@macro{\@outputpage}{\stepcounter{mn@abspage}}%
54 \newcommand*{\@mn@margintest}{%
Number of the next margin note at this page.
55 \expandafter\ifx\csname @mn@thispage\endcsname\@empty
56 \gdef\@mn@atthispage{1}%
57 \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
58 \begingroup
59 \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
60 \xdef\@mn@atthispage{\the\@tempcnta}%
61 \endgroup
62 \else
63 \gdef\@mn@atthispage{1}%
64 \fi
65 \fi
66 \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.

```

67 \let\@mn@currpage\relax
68 \let\@mn@currxpos\relax
69 \if@mn@pdfmode
70   \pdfsavepos
71   \protected@write\@auxout{\let\themn@abspage\relax}{%
72     \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
73       {\themn@abspage}{\noexpand\number\pdfastxpos sp}}}%
74   }%
75 \else
76   \protected@write\@auxout{\let\themn@abspage\relax}{%
77     \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
78       {\themn@abspage}{}}}%
79   }%
80 \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.

```

81 \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.

```

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

```

```

111 \else
112 \if@mn@verbose
113 \PackageInfo{marginnote}{right page because not two side mode}%
114 \fi
115 \@tempwattrue
116 \fi
117 \fi
118 }

```

```

\marginnote Command \marginnote is the main macro of the package. The others are helpers
\@mn@marginnote to manage the optional arguments.
\@mn@@marginnote 119 \newcommand*{\marginnote}{%
\@mn@@@marginnote 120 \@dblarg\@mn@marginnote
121 }
122 \newcommand{\@mn@marginnote}[2][]{%
123 \ifhmode
124 \@bsphack
125 \begingroup
126 \ifdim\@savsk>\z@\else
127 \def\:@@xifnch\expandafter\def\:{ \futurelet\@let@token\@ifnch}%
128 \fi
129 \else
130 \begingroup
131 \fi
132 \ifnextchar [{\@mn@@marginnote[{#1}]{#2}}{\@mn@@marginnote[{#1}]{#2}[\z@]}%
133 }
134 \newcommand{\@mn@@marginnote}{%
135 \def\@mn@@marginnote[#1]#2[#3]{%
136 \endgroup

```

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

```

137 \ifhmode
138 \@mn@@marginnote[{#1}]{#2}[\z@]{#3}%
139 \@esphack
140 \else
141 \@mn@@marginnote[{#1}]{#2}[\z@]{#3}%
142 \fi
143 }
144 \newcommand{\@mn@@marginnote}{%
145 \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.

```

146 \begingroup

```

```

147 \ifmode\strut\let\@tempa\mn@vadjust\else
148 \if@inlabel\leavevmode\fi
149 \ifhmode\strut\let\@tempa\mn@vadjust\else\let\@tempa\mn@vlap\fi
150 \fi
151 \@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 places to the margin.

```

152 \vbox to\z@{%
153 \vss
154 \@mn@margintest
155 \if@reversemargin\if@tempswa
156 \@tempswafalse
157 \else
158 \@tempswatrue
159 \fi\fi
160 \if@tempswa
161 \rlap{%

```

If \@mn@currpos 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.

```

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

```



```

190          \fi
191          \else
192            \kern\oddsidemargin
193          \fi
194          \kern 1in
195        \endgroup
196      \fi
197    \fi
198    \kern\marginnotetextwidth\kern\marginparsep
199    \vbox to\z@{\kern\marginnotevadjust\kern #3
200      \vbox to\z@{%
201        \hsize\marginparwidth

```

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

```

202      \kern-\parskip
203      \marginfont\raggedrightmarginnote\hspace{\z@}\strut#2\endgraf
204      \vss}%
205    \vss}%
206  }%
207 \else
208   \llap{%
209     \vbox to\z@{\kern #3
210       \vbox to\z@{%
211         \hsize\marginparwidth

```

Same like above for left margins.

```

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

```

```

237         \begingroup
238         \kern\@mn@currxpos
239         \if@twoside\ifodd\@mn@currpage\relax
240             \kern-\oddsidemargin
241         \else
242             \kern-\evensidemargin
243         \fi
244     \else
245         \kern-\oddsidemargin
246     \fi
247     \kern-1in
248 \endgroup
249 \fi
250 \fi
251 \kern\marginparsep
252 }%
253 \fi
254 }%
255 }%
256 \endgroup
257 }

```

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

```

258 \newcommand*{\marginnoterightadjust}{}
259 \newcommand*{\marginnoteleftadjust}{}
260 \let\marginnoterightadjust\z@
261 \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.

```

262 \newcommand*{\marginnotevadjust}{}
263 \let\marginnotevadjust\z@

```

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

```

264 \newcommand{\mn@vlap}[1]{%
265   \setbox\@tempboxa\vbox to \ht\strutbox{#1\vss}%
266   \box\@tempboxa\vskip-\baselineskip
267 }

```

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

```

268 \newcommand{\mn@zbox}[1]{%
269   \bgroup
270   \setbox\@tempboxa\vbox{#1}%
271   \ht\@tempboxa\ht\strutbox
272   \dp\@tempboxa\dp\strutbox
273   \wd\@tempboxa\z@

```

```

274 \box\@tempboxa
275 \egroup
276 }

```

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

```

277 \newcommand{\mn@vadjust}[1]{%
278 \mn@zbox{\kern-\parskip
279 \leavevmode\vadjust{#1}%
280 \kern\parskip
281 }%
282 }

```

`\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`.

`\raggedrightmarginnote`

```

283 \providecommand*\marginfont{}
284 \newcommand*\raggedleftmarginnote{\raggedleft}
285 \newcommand*\raggedrightmarginnote{\raggedright}

```

Change History

v1.0a		<code>\@mn@margintest</code> : new PDF mode feature 5
General: Example to macros		<code>\if@mn@pdfmode</code> : new switch 4
<code>\raggedleftmarginnote</code> and		<code>\marginnotetextwidth</code> : new macro 5
<code>\raggedrightmarginnote</code> at documentation fixed [thanks to Susumu Tanimura]. 2	v1.1a	<code>\if@mn@pdfmode</code> : PDF _T E _X since 1.40 allows <code>\pdfsavepos</code> in DVI mode too 4
<code>\marginfont</code> : Use <code>\providecommand</code> to define it. 11	v1.1b	<code>\@mn@@@marginnote</code> : use <code>\mn@vadjust</code> instead of <code>\vadjust</code> 7
v1.0b		<code>\if@mn@pdfmode</code> : if level fixed 4
General: spelling fixes 1		<code>\mn@vadjust</code> : new (internal) 11
v1.1		<code>\mn@zbox</code> : new (internal) 10
<code>\@mn@@@marginnote</code> : new PDF mode feature 7		
<code>\@mn@currpage</code> : new (internal) . . . 5		
<code>\@mn@currxpos</code> : new (internal) . . . 5		