

Non-Floating Margin Notes with `marginnote` Package*

Markus Kohm[†]

2008/11/10

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 v1.1a, last revised 2008/11/10.

[†]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 `\pdflastxpos`). 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!

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@verbostrue
2 \DeclareOption{verbose}{\@mn@verbostrue}
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 PDF_T_EX and which version of PDF_T_EX is used. With PDF_T_EX the horizontal output position may be detected using `\pdfsavepos` and `\pdflastxpos`. So the relative position of the margin may be calculated. Without PDF_T_EX 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
11    \expandafter\ifx\csname pdftexversion\endcsname\relax
12      \begingroup\expandafter\expandafter\expandafter\endgroup
13      \expandafter\ifx\csname pdfoutput\endcsname\relax\else
14        \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi
15      \fi
16    \else
17      \ifnum \pdftexversion<140
18        \begingroup\expandafter\expandafter\expandafter\endgroup
19        \expandafter\ifx\csname pdfoutput\endcsname\relax\else
20          \ifcase\pdfoutput\else\@mn@pdfmodetrue\fi
21        \fi
22      \else
23        \@mn@pdfmodetrue
24      \fi
25    \fi
26  \if@mn@verbose
27    \if@mn@pdfmode
28      \PackageInfo{marginnote}{%
29        \string\pdfoutput\space not 0 and \string\pdflastxpos\space
30        available.\MessageBreak
31        Extended position detection mode activated\@gobble
32      }%
33    \else
34      \PackageInfo{marginnote}{%
35        either \string\pdflastxpos\space or \string\pdfoutput\space not
36        available.\MessageBreak
37        or \string\pdfoutput\space set to 0.\MessageBreak
38        Extended position detection mode deactivated\@gobble
39      }%
40    \fi
41  \fi
42 }
```

`\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.

```
43 \newcommand*{\marginnotetextwidth}{%
44 \let\marginnotetextwidth\textwidth
45 \AtBeginDocument{\if@mn@pdfmode\edef\marginnotetextwidth{\the\textwidth}\fi}
```

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_{TEX} 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!

```
46 \newcommand*{\@mn@thispage}{%
47 \newcommand*{\@mn@currpage}{%
48 \newcommand*{\@mn@currxpos}{%
49 \newcounter{mn@abspage}
50 \AtBeginDocument{\setcounter{mn@abspage}{1}}%
51 \g@addto@macro\@outputpage{\stepcounter{mn@abspage}}%
52 \newcommand*{\@mn@margintest}{%
```

Number of the next margin note at this page.

```
53 \expandafter\ifx\csname @mn@thispage\endcsname\@empty
54 \gdef\@mn@atthispage{1}%
55 \else\expandafter\ifnum \@mn@thispage=\value{mn@abspage}%
56 \begingroup
57 \@tempcnta\@mn@atthispage\advance\@tempcnta by \@ne
58 \xdef\@mn@atthispage{\the\@tempcnta}%
59 \endgroup
60 \else
61 \gdef\@mn@atthispage{1}%
62 \fi
63 \fi
64 \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.

```
65 \let\@mn@currpage\relax
66 \let\@mn@currxpos\relax
67 \if@mn@pdfmode
68 \pdfsavepos
69 \protected@write\@auxout{\let\themn@abspage\relax}{%
70 \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
```

```

71      {\themn@abspage}{\noexpand\number\pdflastxpos sp}}%
72    }%
73  \else
74    \protected@write\@auxout{\let\themn@abspage\relax}{%
75      \string\newmarginnote{note.\@mn@thispage.\@mn@atthispage}{%
76        {\themn@abspage}{}}}%
77    }%
78  \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.

```

79  \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.

```

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

```

```

115 \fi
116 }

\marginnote Command \marginnote is the main macro of the package. The others are helpers
\@mn@marginnote to manage the optional arguments.
\@mn@@marginnote 117 \newcommand*{\marginnote}{%
\@mn@@@marginnote 118 \@dblarg\@mn@marginnote
119 }
120 \newcommand{\@mn@marginnote}[2][]{%
121 \ifhmode
122 \@bsphack
123 \begingroup
124 \ifdim\@savsk>z\@else
125 \def\@xifnch\expandafter\def\@ { \futurelet\@let@token\@ifnch}%
126 \fi
127 \else
128 \begingroup
129 \fi
130 \@ifnextchar [\@mn@@marginnote[{#1}]{#2}]{\@mn@@marginnote[{#1}]{#2}[\z@]}%
131 }
132 \newcommand{\@mn@@marginnote}{%
133 \def\@mn@@marginnote[#1]#2[#3]{%
134 \endgroup

```

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

```

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

```

144 \begingroup
145 \ifmmode\strut\let\@tempa\vadjust\else
146 \if@inlabel\leavevmode\fi
147 \ifhmode\strut\let\@tempa\vadjust\else\let\@tempa\mn@vlap\fi
148 \fi
149 \@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.

```

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

```

160      \ifx\@mn@currpos\relax
161      \kern\marginnoterightadjust
162      \if@mn@verbose
163      \PackageInfo{marginnote}{%
164      xpos not known,\MessageBreak
165      using \string\marginnoterightadjust}%
166      \fi
167      \else\ifx\@mn@currpos\empty
168      \kern\marginnoterightadjust
169      \if@mn@verbose
170      \PackageInfo{marginnote}{%
171      xpos not known,\MessageBreak
172      using \string\marginnoterightadjust}%
173      \fi
174      \else
175      \if@mn@verbose
176      \PackageInfo{marginnote}{%
177      xpos seems to be \@mn@currpos,\MessageBreak
178      \string\marginnoterightadjust
179      \space ignored}%
180      \fi
181      \begingroup
182      \setlength{\@tempdima}{\@mn@currpos}%
183      \kern-\@tempdima
184      \if@twoside\ifodd\@mn@currpage\relax
185      \kern\oddsidemargin
186      \else
187      \kern\evensidemargin
188      \fi
189      \else
190      \kern\oddsidemargin
191      \fi
192      \kern 1in
193      \endgroup

```



```

194         \fi
195     \fi
196     \kern\marginnotetextwidth\kern\marginparsep
197     \vbox to\z@{\kern\marginnotevadjust\kern #3
198         \vbox to\z@{%
199             \hsize\marginparwidth

```

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

```

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

```

Same like above for left margins.

```

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

```

```

241             \fi
242         \else
243             \kern-\oddsidemargin
244         \fi
245         \kern-1in
246     \endgroup
247     \fi
248     \fi
249     \kern\marginparsep
250 }%
251 \fi
252 }%
253 }%
254 \endgroup
255 }

```

`\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.

```

256 \newcommand*{\marginnoterightadjust}{}
257 \newcommand*{\marginnoteleftadjust}{}
258 \let\marginnoterightadjust\z@
259 \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.

```

260 \newcommand*{\marginnotevadjust}{}
261 \let\marginnotevadjust\z@

```

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

```

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

```

`\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`.

```

266 \providecommand*{\marginfont}{}
267 \newcommand*{\raggedleftmarginnote}{\raggedleft}
268 \newcommand*{\raggedrightmarginnote}{\raggedright}

```

Change History

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