

# Das minipage-marginpar-Package – Randnotizen auch in Minipages\*

Paul Ebermann<sup>†</sup>

7. April 2008

## Zusammenfassung

Innerhalb von Boxen wie etwa einer `minipage`-Umgebung sind bekanntlich `\marginpar`-Befehle nicht erlaubt – dieses Package hat eine (Teil-)Lösung.

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Benutzerdoku</b>	<b>1</b>
2.1	Probleme/Nachteile . . . . .	2
<b>3</b>	<b>Implementation</b>	<b>2</b>
3.1	Rückwärtskompatibilitäts-Datei . . . . .	2
3.2	Vorbereitungen . . . . .	3
3.3	Interne Kommandos . . . . .	3
3.4	Neue <code>minipage</code> -Umgebung . . . . .	4
3.5	Ende . . . . .	5
<b>4</b>	<b>Liste der Änderungen</b>	<b>5</b>
<b>5</b>	<b>Index</b>	<b>5</b>

## 1 Einleitung

Innerhalb von Boxen wie etwa einer `minipage`-Umgebung sind `\marginpar`-Befehle nicht erlaubt, das heißt, sie bewirken nichts außer einer Fehlermeldung („Float(s) lost“).

Das Paket `marginnote`<sup>1</sup> von Markus Kohm umgeht das, indem nicht-gleitende Marginalien bereitgestellt werden.

Dieses Paket verfolgt einen anderen Ansatz: Es wird ein Mechanismus bereitgestellt, mit dem die `\marginpar`-Befehle abgefangen und dann außerhalb dieser Box ausgeführt werden können.

---

\*Dieses Dokument gehört zu `minipage-marginpar` v0.3, vom 2008/04/07.

<sup>†</sup>Paul-Ebermann@gmx.de

<sup>1</sup>auf CTAN unter `macros/latex/contrib/marginnote/`

## 2 Benutzerdoku

`minipagewithmarginpars` Diese Umgebung funktioniert wie die `minipage`-Umgebung aus dem L<sup>A</sup>T<sub>E</sub>X-Kernel, mit dem Unterschied, dass in ihr vorkommende `\marginpar`-Befehle erst am Ende der Umgebung, nach der Minipage selbst, ausgeführt werden.

Hier ein Beispiel:

```
Text am Anfang.      \begin{minipagewithmarginpars}{4cm}
                    Text am Anfang\marginpar{Bla}.

                    \vspace{1.5cm}
                    Text am Ende\marginpar[links]{rechts}.
Text am Ende.      \end{minipagewithmarginpars}
```

Die Randnotizen werden in der durch die (hier zwei, der Code ist auch eine) Minipages gebildeten Zeile abgesetzt – dabei ist die Default-Ausrichtung hier `[t]` anstatt `[c]`.

### 2.1 Probleme/Nachteile

- Die Marginalien werden alle ab der selben Zeile (jeweils etwas nach unten verschoben) gesetzt, nicht entsprechend der Zeile (in der Minipage), in der der `\marginpar`-Befehl vorkam.

Dies ist prinzipbedingt, ich habe keine Idee, was man dagegen tun könnte. (Je nach Problem gibt marginnote wohl bessere Ergebnisse.)

- Wird die `minipagewithmarginpars` innerhalb einer weiteren Box verwendet (z.B. um noch einen Rahmen zu setzen), funktioniert es weiterhin nicht, da ja dort ebenfalls keine `\marginpars` erlaubt sind.

Hierfür gibt es eine Lösung – die gesicherten `\marginpars` müssen einfach erst nach der äußersten Box (die hoffentlich in einer horizontalen Liste ist) ausgeführt werden. Dazu kann man sich analog zu `minipagewithmarginpars` eine entsprechende Umgebung definieren, Details sind in Abschnitt 3.4 im Implementations-Teil nachzulesen.

- Befinden sich die `\marginpar`-Befehle innerhalb einer der AMS-Mathe-Umgebungen (wie etwa `align`, `gather` und ihre \*-Varianten), so kann es vorkommen, dass die Marginalien doppelt auftauchen. Dies liegt daran, dass diese Umgebungen ihren Inhalt mehrfach auswerten – und jedesmal wird dann der Inhalt gespeichert. Dieses Problem betrifft auch andere Makros, die ähnlich vorgehen.

Ich halte dies für einen Bug (in `minipage-marginpar`), habe aber noch keine Idee, was ich dagegen tun könnte. Wer eine Idee hat, immer her damit.

## 3 Implementation

### 3.1 Rückwärtskompatibilitäts-Datei

CTAN hat ja inzwischen seine 8-Buchstaben-pro-Dateiname-Regel aufgehoben. Genauer, sie wurde zwischen meinem ersten Versuch, ein (anderes) Paket mit zu langem Dateinamen hochzuladen, und dem ersten Hochladen dieses Paketes

geändert, so dass ich dieses Package noch `mpgmpar` genannt habe, es bei CTAN dann aber in einem Verzeichnis `minipage-marginpar` gelandet ist, und auch unter diesem Namen im CTAN-Katalog zu finden ist. Für Version 0.3 habe ich mich daher entschieden, auch meine Package-Dateien entsprechend zu benennen. Hier aber noch für die Rückwärtskompatibilität eine passende `mpgmpar.sty`:

```
1 (*mpgmpar)
2 \RequirePackage{minipage-marginpar}
3 \mpgmpar
```

Und jetzt geht das eigentliche Package los.

```
4 (*package)
```

## 3.2 Vorbereitungen

`\mpgmpar@savedmargins` Dieses Makro ist einfach nur ein „Behälter“ für die aufgesparten `\marginpar`-Befehle. Wir definieren es hier (leer) mittels `\newcommand`, um bei Konflikten eine Fehlermeldung zu erhalten.

```
5 \newcommand*\mpgmpar@savedmargins{}%
```

`\mpgmpar@dumy` Diese Kontrollsequenz wird nur verwendet, um einen nicht vorhandenen Parameter erkennen zu können. (Wir definieren es zunächst als Makro, um Konflikte zu erkennen, lassen es nachher aber gleich `\relax` sein.)

```
6 \newcommand*\mpgmpar@dumy{}%
7 \let \mpgmpar@dumy = \relax
```

## 3.3 Interne Kommandos

Unsere beiden Makros `\mpgmpar@savemarginpars` (am Anfang eines Bereiches) und `\mpgmpar@restoremarginpars` (am Ende) machen die eigentliche Arbeit und können auch für die Definition eigener *Box-Making*-Umgebungen verwendet werden, für eine Anleitung dafür siehe Abschnitt 3.4.

`\mpgmpar@savemarginpars` Dieses Makro leitet einen Bereich ein, in dem `\marginpars` aufgespart werden (er geht bis zum Ende der aktuellen Gruppe).

```
8 \newcommand*\mpgmpar@savemarginpars{}%
```

Wir definieren `\marginpar` neu.

`\marginpar` Es hat wie das Original-`\marginpar` einen optionalen und einen verpflichtenden Parameter. (Um den Fall, dass der optionale Parameter angegeben wurde, von dem Fall der Nichtangabe zu unterscheiden, nehmen wir als Default `\mpgmpar@dumy` und vergleichen nachher damit.)

```
9 \renewcommand*\marginpar}[2] [\mpgmpar@dumy]%
10 {%
```

Das `\@bsphack` und das dazugehörige `\@esphack` am Ende sind Kernel-Kommandos, welche (zusammen) dafür sorgen, dass an der Stelle der Verwendung keine Spur des `\marginpar`-Aufrufs bleibt.

```
11 \@bsphack
```

Jetzt der Vergleich ... bis zu Version 0.1 hatte ich den mit

```
\ifthenelse{\equal{\mpgmpar@dumy}{\#1}}{...}
```

aus `ifthen` gemacht, aber das hatte Nebenwirkungen (da für den Vergleich `##1` expandiert wurde). Daher direkt mit `\ifx`. In Version 0.2 war dieser `\ifx`-Vergleich kaputt (ergab immer `false`), wodurch `\marginpar{bla}` nur auf rechten Seiten gedruckt wurde. Jetzt (0.3) sollte es gehen.

```
12 \def\temp@a{##1}%
13 \def\temp@b{\mpgmpar@dummy}%
14 \ifx\temp@b\temp@a%
```

Dann kopieren wir (global) einfach den `\marginpar`-Aufruf an das Ende unseres „Speicher-Makros“ `\mpgmpar@savedmargins`.

```
15 \g@addto@macro{\mpgmpar@savedmargins}{%
16 \marginpar{##2}}%
17 \else%
18 \g@addto@macro{\mpgmpar@savedmargins}{%
19 \marginpar[##1]{##2}}%
20 \fi%
21 \@ignorefalse
22 \@esphack
23 }%
24 }%
```

Das `\@bsphack-\@esphack`-Paar ist hier vorhanden, damit sich unser modifiziertes `\marginpar` bezüglich umrundender Leerzeichen o.ä. genauso verhält wie das Original-`\marginpar`.

`\mpgmpar@restoremarginpars` Dieses Makro führt die gespeicherten `\marginpar`-Befehle aus und leert dann die Liste.

```
25 \newcommand*{\mpgmpar@restoremarginpars}{%
```

Zuerst merken wir uns die Liste im Makro `\@tempa` (das ist für derartige Sachen gedacht), dann löschen wir (global) `\mpgmpar@savedmargins`.

```
26 \let \@tempa = \mpgmpar@savedmargins
27 \global\let \mpgmpar@savedmargins = \empty
```

Die eben kopierte Liste führen wir nun aus (falls wir jetzt noch in einer äußeren Umgebung mit unserem Spezial-`\marginpar` sind, wird die Liste dadurch neu angelegt, andernfalls werden die Randnotizen ausgegeben), und löschen dann die Kopie.

```
28 \@tempa
29 \let \@tempa = \undefined
30 }%
```

### 3.4 Neue minipage-Umgebung

Diese Umgebung dient als Beispiel für die Erstellung derartiger Umgebungen mit Hilfe von `\mpgmpar@savemarginpars` und `\mpgmpar@restoremarginpars`. Das Wesentliche dabei ist: `\mpgmpar@savemarginpars` sollte innerhalb einer Gruppe aufgerufen werden (meist nahe am Anfang), `\mpgmpar@restoremarginpars` nach dem Ende dieser Gruppe (an der Stelle, wo die Randnotizen erscheinen sollen).

Mit etwas Eigenarbeit dürfte es auch ohne eine (weitere) Gruppe klappen. Dafür muss man am Anfang `\marginpar` mit `\let` sichern und am Ende wiederherstellen:

```

\let \savedmarginpar = \marginpar
\mpgmpar@savemarginpars
...
\let \marginpar = \savedmarginpar
\mpgmpar@restoremarginpars

```

Das ist aber nicht von mir getestet, also ohne Garantie. Und es ist eher nicht robust gegenüber Verschachtelungen derartiger Konstrukte.

**minipagewithmarginpars** Hier nun unserer neue Minipage-Umgebung. Sie hat einen optionalen (vertikale Ausrichtung – Default ist `t`) und einen verpflichtenden (Breite) Parameter.

```
31 \newenvironment*{minipagewithmarginpars}[2][t]{%
```

Die Implementation ist einfach: Wir beginnen zunächst die Original-minipage-Umgebung (mit den selben Parametern) und innerhalb davon rufen wir unser Makro `\mpgmpar@savemarginpars` auf.

```
32   \begin{minipage}[#1]{#2}%
33     \mpgmpar@savemarginpars
34 } {%
```

Am Ende beenden wir zunächst die Minipage (wodurch `\marginpar` seine Original-Bedeutung wiedererlangt), und rufen dann `\mpgmpar@restoremarginpars` auf.

```
35   \end{minipage}%
36   \mpgmpar@restoremarginpars
37 }%
```

### 3.5 Ende

... Das war es.

```
38 \endinput
39 \endpackage
```

## 4 Liste der Änderungen

v0.0		<code>\ifx</code> statt <code>\ifthenelse</code> . . . . .	3	
	Allgemein: Erste Fassung . . . . .	1	v0.3	
v0.1	Allgemein: Erste veröffentlichte Fassung . . . . .	1	Allgemein: Neuer Name des Packages: <code>minipage-marginpar</code> statt <code>minipage-marginpar</code> . . . . .	1
v0.2	Allgemein: <code>ifthen</code> wird nicht mehr benötigt. . . . .	3	<code>\marginpar</code> : Fehler im Vergleich gefunden, jetzt korrigiert durch weitere Indirektion beim Vergleich. . . . .	3
	<code>\marginpar</code> : Vergleich jetzt mit			

## 5 Index

Schräggedruckte Nummern verweisen auf die Seite, auf der der Eintrag beschrieben ist, unterstrichene Nummern zeigen auf die Zeilennummer der Definition, sonstige Zahlen auf die Zeilennummer einer Verwendung.

	<b>Symbols</b>	<code>\@empty</code> . . . . .	27
<code>\@bsphack</code> . . . . .			11

\@esphack .....	22	<b>I</b>	
\@ignorefalse .....	21	\ifx .....	14
<b>D</b>		<b>M</b>	
\def .....	12, 13	\marginpar .....	<u>9</u>
<b>E</b>		minipagewithmarginpars (environ-	<u>31</u>
\else .....	17	ment) .....	<u>31</u>
environments:		\mpgmpar@dummy .....	<u>6</u> , 9, 13
minipagewithmarginpars .....	<u>31</u>	\mpgmpar@restoremarginpars ...	<u>25</u> , 36
<b>F</b>		\mpgmpar@savemargins <u>5</u> , 15, 18, 26, 27	
\fi .....	20	\mpgmpar@savemarginpars .....	<u>8</u> , 33
<b>G</b>		<b>T</b>	
\g@addto@macro .....	15, 18	\temp@a .....	12, 14
		\temp@b .....	13, 14