

Package `boites`*

Vincent Zoonekynd†

Susan Dittmar (translation of the documentation) ‡

May 6, 2009

Part II English

1 Modifications by VZ, July 1998

May 2009: Original file `README` included

January 2009: Translations of the french comments by Susan Dittmar.

March 1999: Some lines are not taken in correctly (for example those that contain only vertical space before or after an enumeration).

March 1999: comments

There are some bugs, in particular some lengths that are too long, too short, too low, or too fine. If someone knows what causes this, please tell me.

The problem with list environments (itemize, enumerate, etc.) inside the `boite` environment should not longer exist.

Based upon: `eclbkbbox.sty` by Hideki Isozaki, 1992; Date: May 28, 1993.

2 User Documentation

`boites.sty` (boxes that may break across pages)

© 1998–1999 Vincent Zoonekynd <zoonek@math.jussieu.fr>

Distributed under the GNU Public Licence

*v1.0a, 2009/05/06

†zoonek@math.jussieu.fr

‡Susan.Dittmar@gmx.de

2.1 Description

These environments allow page breaks inside framed boxes. They include a few examples (shaded box, box with a wavy line on its side, etc.)

See also `macros/latex/contrib/other/misc/framed.sty`.

2.2 Usage

In the preamble:

```
\usepackage{boites,boites_examples,graphicx}
```

Before using the various environments:

`\bkcounttrue`: the lines will be numbered

`\bkcountfalse`: the lines will not be numbered

Boxed text with a title:

```
\begin{boiteepaisseavecuntitre}  
...  
\end{boiteepaisseavecuntitre}
```

Text with a double vertical line on the left and a number (17, in this example):

```
\begin{boitenumeroteeavecunedoublebarre}{17}  
...  
\end{boitenumeroteeavecunedoublebarre}
```

Text with a wavy line on the left:

```
\begin{boiteavecunelignequiondulesurlecote}  
...  
\end{boiteavecunelignequiondulesurlecote}
```

Shaded box:

```
\begin{boitecoloriee}  
...  
\end{boitecoloriee}
```

If you wish other kinds of boxes, have a look at `boites_examples.sty` and feel free to adapt the macros. Examples for several kinds of boxes are shown at `boites_exemples.pdf`.

2.3 Features

- These environments may be nested.
- They may appear in a multicols environment.
- Floating material, footnotes, marginpars appearing inside them will be lost.

3 Implementation

```
1 \newbox\bk@bxb
2 \newbox\bk@bxa
3 \newif\if@bkcont
4 \newif\if@bkcount
5 \newcount\bk@lcnt
6
7 \def\breakboxskip{2pt}
8 \def\breakboxparindent{1.8em}
```

3.1 parameters that can be modified

```
9 \def\bkvz@before@breakbox{\ifhmode\par\fi\vskip\breakboxskip\relax}
```

What should be put on the left of the text, for example a vertical line for a frame, or a wavy line.

```
10 \def\bkvz@left{\vrule \@width\fbboxrule\hskip\fbboxsep}
```

Analog; what should be put on the right,

```
11 \def\bkvz@right{\hskip\fbboxsep\vrule \@width\fbboxrule}
```

above

```
12 \def\bkvz@top{\hrule\@height\fbboxrule}
```

and below.

```
13 \def\bkvz@bottom{\hrule\@height\fbboxrule}
```

If you modify one of those macros, don't forget to modify also the following. It reduces the value of `\linewidth` by the width of all that will be put on the edges.

```
14 \def\bkvz@set@linewidth{\advance\linewidth -2\fbboxrule
```

```
15 \advance\linewidth -2\fbboxsep}
```

END OF PARAMETERS THAT CAN BE MODIFIED

3.2 The start of the environment

```
16 \def\breakbox{%
```

Start is not necessarily in vertical mode. `\bkvz@before@breakbox` deals with this (if necessary).

```
17 \bkvz@before@breakbox
```

put all in one `\vbox` (`\bk@bxb`)

```
18 \setbox\bk@bxb\vbox\bgroup
```

Inside this `\vbox`, change `\hsize` (and `\linewidth`).

```
19 \bkvz@set@linewidth
```

```
20 \hsize\linewidth
```

I do not know what `\@parboxrestore` does.

```

21 \@parboxrestore

```

Indent if the user so desires.

```

22 \parindent\breakboxparindent\relax}

```

Cut the box

`\@tempdimb`: amount of vertical skip between the first line (`\bk@bxa`) and the rest (`\bk@bxb`)

```

23 \def\bk@split{%

```

Calculate the total height (height + depth) of the box.

```

24 \@tempdimb\ht\bk@bxb % height of original box
25 \advance\@tempdimb\dp\bk@bxb

```

Cut with the help of `\vsplit...to` Opt The height can then be found in `\bk@bxa`, the depth in `\bk@bxb`.

```

26 \setbox\bk@bxa\vsplit\bk@bxb to\z@ % split it

```

A problem arises if the first box has an empty height. It can be given back its initial height via `\vbox{\unvbox...}`

```

27 \setbox\bk@bxa\vbox{\unvbox\bk@bxa}% recover height & depth of \bk@bxa

```

The other problem is to forget the (interline) space between our our two boxes. To regain it, add the height of the two boxes and subtract that from the initial height.

```

28 \setbox\@tempboxa\vbox{\copy\bk@bxa\copy\bk@bxb}% naive concatenation
29 \advance\@tempdimb-\ht\@tempboxa
30 \advance\@tempdimb-\dp\@tempboxa

```

Now, `\@tempdimb` contains the space between the two boxes, which will be used with `\bk@addskipdp`.

```

31 }% gap between two boxes

```

Add `\fboxsep` to the first line

`\@tempdima`: height of the first line (`\bk@bxa`) + `\fboxsep`

```

32 \def\bk@addfsepht{%
33   \setbox\bk@bxa\vbox{\vskip\fboxsep\box\bk@bxa}}

```

This macro is not used anywhere

```

34 \def\bk@addskipht{%
35   \setbox\bk@bxa\vbox{\vskip\@tempdimb\box\bk@bxa}}

```

Add `\fboxsep` to the last line

`\@tempdima`: depth of the first line (`\bk@bxa`) + `\fboxsep`

```

36 \def\bk@addfsepdp{%
37   \@tempdima\dp\bk@bxa
38   \advance\@tempdima\fboxsep
39   \dp\bk@bxa\@tempdima}

```

Add the space that had been lost by `\vsplit...to` Opt

`\@tempdima`: depth of the first line (`\bk@bxa`) + vertical skip

```

40 \def\bk@addskipdp{%
41   \@tempdima\dp\bk@bxa
42   \advance\@tempdima\@tempdimb
43   \dp\bk@bxa\@tempdima}

```

Not all lines are computed, only cells that truly are there. I have taken as criterion a size of minimum 1mm. The same distance can be found further on, in `\bk@line`.

```

44 \def\bkvz@countlines{%
45   \ifdim\wd\bk@bxa>1mm\advance\bk@lcnt\@ne\fi
46 }

    show the line we had cut
47 \def\bk@line{%
48   \hbox to \linewidth{%
49     \ifdim\wd\bk@bxa>1mm
50       \ifbkcount\smash{\llap{\the\bk@lcnt\ }}\fi
51     \fi
52     \bkvz@left
53     \box\bk@bxa

```

Sometimes the box is not big enough for a line (for example, space before an enumeration)

```

54   \hfil
55   \bkvz@right}}

```

The end of the environment

```

56 \def\endbreakbox{%
Close the \vbox (\bk@bxb)
57 \egroup
58 % \ifhmode\par\fi
59 {\noindent

```

Set line count back to one.

```

60   \bk@lcnt 0

```

The boolean we will use in the following loop.

```

61   \@bkconttrue

```

While putting together the boxes, some ajustable lengths are set to zero to avoid undesired vertical space.

```

62   \baselineskip\z@
63   \lineskiplimit\z@
64   \lineskip\z@
65   \vfuzz\maxdimen

```

split the boxes

```

66   \bk@split

```

Add a bit of vertical space (`\fboxsep`) above

```

67   \bk@addfsept

```

Add below the space that had been forgotten due to the use of the `\vsplit` command.

```

68   \bk@addskipdp

```

First of two,

```

69   \ifvoid\bk@bxb

```

In case there's only one line

```

70   \def\bk@fstln{%

```

Add a bit of space (`\fboxsep`) below.

```
71      \bk@addfsepdp
```

Construct the box: the top, the middle (which contains the left and right parts) and the foot.

```
72      \bkvz@countlines
73      \vbox{\bkvz@top\bk@line\bkvz@bottom}}%
```

In case there's more to do.

```
74      \else
75      \def\bk@fstln{%
```

Put in the top

```
76      \bkvz@countlines
77      \vbox{\bkvz@top\bk@line}%
```

??? (If this is removed, it does not work any more.)

```
78      \hfil
```

Continue counting lines

```
79 %      \advance\bk@lcnt\@ne %%%%%%%%%%%%%% Voir \bkvz@countlines
```

Begin of the loop

```
80      \loop
```

Cut out the next bit of the box.

```
81      \bk@split
```

Add the vertical space that has been left out.

```
82      \bk@addskipdp
```

advance number of lines if necessary

```
83      \bkvz@countlines
```

???

```
84      leavevmode
```

If it's the last line

```
85      \ifvoid\bk@bxb
```

Set the boolean that indicates continuing the loop to FALSE.

```
86      \@bkcontfalse
```

Add a bit of vertical space (`\fboxsep`)

```
87      \bk@addfsepdp
```

Add the last line. WHY `\vtop`??? Because the line number has correct height.

```
88      \vtop{\bk@line\bkvz@bottom}%
89      \else % 2,...,(n-1)
90      \bk@line
91      \fi
92      \hfil
```

See `\bkvz@countlines`

```
93 %      \advance\bk@lcnt\@ne
```

```
94      \if@bkcont\repeat}%
```

```
95      \fi
```

```
96      \leavevmode\bk@fstln\par}\vskip\breakboxskip\relax}
```

```
97 \bkcountfalse
```

boites*

Vincent Zoonekynd†

2009/05/06

Troisième partie

Franzaise

1 Modifications par VZ, Juillet 1998

Mars 1999 : Il y a certaines lignes à ne pas numéroter (par exemple, celles qui ne contiennent que des espaces verticaux avant ou après une énumération).

Mars 1999 : commentaires

Il y a quelques bugs, en particulier des traits qui sont trop longs, trop courts, trop épais ou trop fins. Si Quelqu'un sait à quoi c'est dû, qu'il me le dise.

Il ne devrait plus y avoir de problème à cause d'un environnement de type liste (itemize, enumerate, etc.) à l'intérieur des boites.

D'après : eclbkbbox.sty by Hideki Isozaki, 1992 ; Date : May 28, 1993.

2 Guide d'utilisateur

Cette section n'existe malheureusement que dans la version anglaise.

3 Implementation

```
1 \newbox\bk@bxb
2 \newbox\bk@bxa
3 \newif\if@bkcont
4 \newif\ifbkcount
5 \newcount\bk@lcnt
6
7 \def\breakboxskip{2pt}
8 \def\breakboxparindent{1.8em}
```

*v1.0a, 2009/05/06

†zoonek@math.jussieu.fr

3.1 Paramètres modifiables

```
9 \def\bkvz@before@breakbox{\ifhmode\par\fi\vskip\breakboxskip\relax}
```

Ce que l'on met à gauche du texte, par exemple, une ligne verticale pour faire un cadre, ou une ligne qui ondule.

```
10 \def\bkvz@left{\vrule \@width\fbboxrule\hskip\fbboxsep}
```

De même, ce que l'on met à droite,

```
11 \def\bkvz@right{\hskip\fbboxsep\vrule \@width\fbboxrule}
```

en haut

```
12 \def\bkvz@top{\hrule\@height\fbboxrule}
```

ou en bas

```
13 \def\bkvz@bottom{\hrule\@height\fbboxrule}
```

Si vous modifiez l'une de ces macros, il ne faut pas oublier de modifier aussi la suivante, qui change la valeur de `\linewidth` en lui retirant la largeur de tout ce que l'on vient de mettre sur le côté.

```
14 \def\bkvz@set@linewidth{\advance\linewidth -2\fbboxrule}
```

```
15 \advance\linewidth -2\fbboxsep}
```

FIN DES PARAMÈTRES MODIFIABLES

3.2 Le début de l'environnement

```
16 \def\breakbox{%
```

On n'est pas nécessairement en mode vertical. C'est `\bkvz@before@breakbox` qui s'en occupe (ou non).

```
17 \bkvz@before@breakbox
```

on met tout dans une `\vbox` (`\bk@bxb`)

```
18 \setbox\bk@bxb\vbox\bgroup
```

À l'intérieur de cette `\vbox`, on change la valeur de `\hsize` (et aussi `\linewidth`).

```
19 \bkvz@set@linewidth
```

```
20 \hsize\linewidth
```

je ne sais pas ce que fait la commande `\@parboxrestore`.

```
21 \@parboxrestore
```

On indente éventuellement, si l'utilisateur le désire.

```
22 \parindent\breakboxparindent\relax}
```

On coupe la boîte

`\@tempdimb` : amount of vertical skip between the first line (`\bk@bxa`) and the rest (`\bk@bxb`)

```
23 \def\bk@split{%
```

On calcule la hauteur totale (hauteur + profondeur) de la boîte.

```
24 \@tempdimb\ht\bk@bxb % height of original box
```

```
25 \advance\@tempdimb\dp\bk@bxb
```

On coupe, à l'aide de la commande `\vsplit...to 0pt` Le morceau du haut se retrouve dans `\bk@bxa`, celui du bas dans `\bk@bxb`.

```
26 \setbox\bk@bxa\vsplit\bk@bxb to\z@ % split it
```


L'un des problèmes, c'est que la première boîte a une hauteur vide. On peut lui redonner sa hauteur initiale grâce à `\vbox{\unvbox...}`

```
27 \setbox\bk@bxa\vbox{\unvbox\bk@bxa}% recover height & depth of \bk@bxa
```

L'autre problème, c'est que l'on a perdu l'espace (interligne) entre nos deux boîtes. Pour le récupérer, on ajoute la hauteur de ces deux boîtes, et on fait la différence avec la hauteur initiale.

```
28 \setbox\@tempboxa\vbox{\copy\bk@bxa\copy\bk@bxb}% naive concatenation
```

```
29 \advance\@tempdimb-\ht\@tempboxa
```

```
30 \advance\@tempdimb-\dp\@tempboxa
```

Désormais, `\@tempdimb` contient l'espace entre les deux boîtes, que l'on utilisera avec `\bk@addskipdp`.

```
31 }% gap between two boxes
```

Rajouter `\fboxsep` à la première ligne

```
\@tempdima : height of the first line (\bk@bxa) + \fboxsep
```

```
32 \def\bk@addfsepht{%
```

```
33 \setbox\bk@bxa\vbox{\vskip\fboxsep\box\bk@bxa}}
```

Cette macro n'est pas utilisée

```
34 \def\bk@addskiptht{%
```

```
35 \setbox\bk@bxa\vbox{\vskip\@tempdimb\box\bk@bxa}}
```

Rajouter `\fboxsep` à la dernière ligne

```
\@tempdima : depth of the first line (\bk@bxa) + \fboxsep
```

```
36 \def\bk@addfsepdp{%
```

```
37 \@tempdima\dp\bk@bxa
```

```
38 \advance\@tempdima\fboxsep
```

```
39 \dp\bk@bxa\@tempdima}
```

Rajouter l'espace qui a été perdu par `\vsplit... to Opt`

```
\@tempdima : depth of the first line (\bk@bxa) + vertical skip
```

```
40 \def\bk@addskipdp{%
```

```
41 \@tempdima\dp\bk@bxa
```

```
42 \advance\@tempdima\@tempdimb
```

```
43 \dp\bk@bxa\@tempdima}
```

On ne compte pas toutes les lignes, mais uniquement celles qui en sont vraiment. J'ai pris comme critère une largeur supérieure à 1mm. La même distance se retrouve un peu plus loin, dans `\bk@line`.

```
44 \def\bkvz@countlines{%
```

```
45 \ifdim\wd\bk@bxa>1mm\advance\bk@lcnt\@ne\fi
```

```
46 }
```

Afficher la ligne que l'on vient de couper

```
47 \def\bk@line{%
```

```
48 \hbox to \linewidth{%
```

```
49 \ifdim\wd\bk@bxa>1mm
```

```
50 \ifbkcoun\smash{\llap{\the\bk@lcnt\ }}\fi
```

```
51 \fi
```

```
52 \bkvz@left
```

```
53 \box\bk@bxa
```

Il arrive que la boite ne soit pas aussi large que la ligne (par exemple, espace avant une énumération)

```
54     \hfil
55     \bkvz@right}}
```

La fin de l'environnement

```
56 \def\endbreakbox{%
On ferme la \vbox (\bk@bxb)
57 \egroup
58 % \ifhmode\par\fi
59 {\noindent
```

On remet le compteur de lignes à un.

```
60     \bk@lcnt 0
```

Le booléen que nous allons utiliser dans la boucle plus loin.

```
61     \@bkconttrue
```

Comme on va empiler des boites, on met certains ressorts à zéro, pour éviter les espaces verticaux non désirés.

```
62     \baselineskip\z@
63     \lineskiplimit\z@
64     \lineskip\z@
65     \vfuzz\maxdimen
```

On coupe la boite

```
66     \bk@split
```

On ajoute un peu d'espace vertical (\fboxsep) au dessus

```
67     \bk@addfsepht
```

On ajoute en dessous l'espace qui avait été perdu par la commande \vsplit.

```
68     \bk@addskipdp
```

De deux choses l'une,

```
69     \ifvoid\bk@bxb
```

Soit, il n'y a qu'une ligne

```
70     \def\bk@fstln{%
```

On rajoute un peu d'espace (\fboxsep) en dessous.

```
71     \bk@addfsepdp
```

On construit la boite : le haut, le milieu (qui contient la gauche et la droite) et le bas.

```
72     \bkvz@countlines
73     \vbox{\bkvz@top\bk@line\bkvz@bottom}}%
```

Soit, il y en a plusieurs.

```
74     \else
75     \def\bk@fstln{%
```

On met le haut

```
76     \bkvz@countlines
77     \vbox{\bkvz@top\bk@line}%
```

??? (Si on l'enlève, ça ne marche plus.)

```
78     \hfil
```

On commence à compter les lignes

```
79 % \advance\bk@lcnt\@ne %%%%%%%%%%% Voir \bkvz@countlines
```

Début de la boucle

```
80 \loop
```

On coupe ce qui reste de la boîte.

```
81 \bk@split
```

On rajoute l'espace vertical qui a été perdu.

```
82 \bk@addskipdp
```

Éventuellement, on augmente le numéro de la ligne

```
83 \bkvz@countlines
```

???

```
84 \leavevmode
```

S'il s'agit de la dernière ligne

```
85 \ifvoid\bk@bxb
```

On met le booléen indiquant que la boucle doit se poursuivre à FAUX.

```
86 \@bkcountfalse
```

On met un peu d'espace vertical (`\fboxsep`)

```
87 \bk@addfsepdp
```

Envoie la dernière ligne. POURQUOI `\vtop` ??? Pour que l'éventuel numéro de ligne soit à la bonne hauteur.

```
88 \vtop{\bk@line\bkvz@bottom}%
89 \else % 2,...,(n-1)
90 \bk@line
91 \fi
92 \hfil
```

Voir `\bkvz@countlines`

```
93 % \advance\bk@lcnt\@ne
94 \if@bkcount\repeat}%
95 \fi
96 \leavevmode\bk@fstln\par}\vskip\breakboxskip\relax}
97 \bkcountfalse
```