

# Dashed and layered boxes

Reuben Thomas  
rrt@sc3d.org

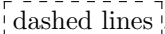

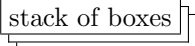
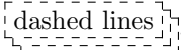
11th December 2001

## Abstract

`dashbox` provides new commands similar to `\framebox` and `\fbox` to typeset dashed and layered boxes.

## 1 User interface

The following commands are provided:

- `\dbox` `\dbox{text}` works like `\fbox`, but the box is drawn with .
- `\dashbox` `\dashbox[width] [pos]{text}` works like `\framebox`, but the box is drawn with .
- `\lbox` `\lbox[layers]{text}` draws a  around its contents, with the number of layers given by the first parameter (default 2).
- `\dlbox` `\dlbox[layers]{text}` works like `\lbox`, but the boxes are drawn with .

The following style parameters are available:

- `\dashlength` `\dashlength` gives the length of a dash plus the following gap. The default is 6pt.
- `\dashdash` `\dashdash` gives the length of a dash. The default is 3pt.
- `\layersize` `\layersize` gives the protrusion of each layer below the previous one. The default is `\dashdash`.

The following standard parameters are also observed:

- `\fboxrule` `\fboxrule` gives the width of the dashes.
- `\fboxsep` `\fboxsep` gives the separation between the box and text inside it.

## 2 Implementation

### 2.1 Preliminaries

Make sure we've got what we need, and announce the package.

```
1 ⟨*package⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{dashbox}
4   [2001/12/11 v1.14 Dashed and layered boxes]
5 \RequirePackage{calc}
6 \RequirePackage{ifthen}
```

### 2.2 Style parameters

Define and give the default values of the style parameters.

```
\dashlength
7 \newlength{\dashlength} \setlength{\dashlength}{6pt}

\dashdash
8 \newlength{\dashdash} \setlength{\dashdash}{3pt}

\layersize
9 \newlength{\layersize} \setlength{\layersize}{\dashdash}
```

### 2.3 Dashes

We need two new commands for drawing horizontal and vertical dashes.

`\hd@shrule` `\hd@shrule` takes one argument, the rule's width. The thickness of the dash is given by `\fboxrule`.

```
10 \newcommand{\hd@shrule}[1]{%
11   \hbox to #1%
12     {\vrule height \fboxrule width \dashdash%
13       \cleaders\hbox to \dashlength%
14         {\hfill\rule{\dashdash}{\fboxrule}\hfill}\hfill%
15       \ifthenelse{\lengthtest{#1 > 2\dashdash}}%
16         {\vrule height \fboxrule width \dashdash}{}}%
17   }
```

`\vd@shrule` `\vd@shrule` takes one argument, the rule's height. The thickness of the dash is given by `\fboxrule`.

```
18 \newcommand{\vd@shrule}[1]{%
19   \vbox to #1%
20     {\hrule height \dashdash width \fboxrule%
21       \cleaders\vbox to \dashlength%
22         {\vfill\rule{\fboxrule}{\dashdash}\vfill}\vfill%
23       \ifthenelse{\lengthtest{#1 > 2\dashdash}}%
24         {\hrule height \dashdash width \fboxrule}{}}%
25   }
```

## 2.4 Dashed boxes

A private save box and some lengths are defined. `\d@ashedsavebox` is a box to hold the contents of a dashed box. `\d@shedboxwidth` is the box's width, and `\d@shedboxtotalheight` is the height plus the depth.

```
\d@shedsavebox
26 \newsavebox{\d@shedsavebox}

\d@shedboxwidth
27 \newlength{\d@shedboxwidth}

\d@shedboxtotalheight
28 \newlength{\d@shedboxtotalheight}

\m@kedashbox \m@kedashbox is where the work is actually done. It puts the box together piece
by piece. It requires \d@shedboxwidth to be set by its caller.
29 \newcommand{\m@kedashbox}{%
    The height plus depth of the box is calculated.
30     \setlength{\d@shedboxtotalheight}{%
31         {\dp\d@shedsavebox+\ht\d@shedsavebox+\fboxsep*2+\fboxrule*2}%
    The box is raised an appropriate amount, and drawn in a b-aligned parbox.
32     \raisebox{-\fboxrule-\fboxsep-\dp\d@shedsavebox}{%
33         \parbox[b]{\d@shedboxwidth}{%
    Inter-line and inter-paragraph skip are disabled.
34         \offinterlineskip%
35         \parskip=0pt%
    The top line is drawn; the kern makes the left and right sides line up properly.
36         \hd@shrulerule{\d@shedboxwidth}%
37         \kern-\fboxrule%
38         \par%
    The left-hand side is now drawn, in a parbox of the correct width.
39         \parbox{\fboxrule}{\vd@shrulerule{\d@shedboxtotalheight}}%
    Now the inside of the box is set in a parbox, with \fboxsep space top and
    bottom. The kerns add \fboxsep space at the left and right.
40         \kern\fboxsep%
41         \parbox{\wd\d@shedsavebox}{%
42             {\vspace{\fboxsep}\usebox{\d@shedsavebox}\vspace{\fboxsep}}%
43         \kern\fboxsep%
```

The right-hand side is drawn just like the left-hand side, and the bottom just like the top.

```

44     \parbox{\fboxrule}{\vd@shrulerule{\d@shedboxtotalheight}}%
45     \par%
46     \kern-\fboxrule%
47     \hd@shrulerule{\d@shedboxwidth}}%
48   }}

```

`\dbox` `\dbox` is just a wrapper around `\m@kedashbox` which saves its argument and then calculates the width according to that of its argument.

```

49 \newcommand{\dbox}[1]{%
50   \sbox{\d@shedsavebox}{#1}%
51   \setlength{\d@shedboxwidth}{\wd\d@shedsavebox+\fboxsep*2+\fboxrule*2}%
52   \m@kedashbox}

```

`\dashbox` The code for `\dashbox` is partly taken from that for `\framebox`. Depending on whether any optional arguments are given, it either simply calls `\dbox`, or sets the width to that given and does the typesetting via `\savebox` and `\m@kedashbox`.

```

53 \def\dashbox{\@ifnextchar[\d@dashbox\dbox}
54 \def\d@dashbox[#1]{\@ifnextchar[{\d@idashbox[#1]}{\d@idashbox[#1][c]}}
55 \long\def\d@idashbox[#1][#2]#3%
56   {\setlength{\d@shedboxwidth}{#1}%
57   \savebox{\d@shedsavebox}[#1-\fboxsep*2-\fboxrule*2][#2][#3]%
58   \m@kedashbox}

```

## 2.5 Layers

Another series of private variables are required for layers: `\l@yersavebox` holds the text to be set in a layer, `\l@yerwidth` holds the total width of the layer, `\l@yerboxwidth` the width of the layer box, `\l@yertotalheight` the height plus depth of the layer. `\l@yerlineheight` the lift of the top right-hand line, and `\l@yervoffset` the lift of the layer below the baseline.

```

\l@yersavebox
59 \newsavebox{\l@yersavebox}

\l@yerwidth
60 \newlength{\l@yerwidth}

\l@yerboxwidth
61 \newlength{\l@yerboxwidth}

\l@yertotalheight
62 \newlength{\l@yertotalheight}

\l@yerlineheight
63 \newlength{\l@yerlineheight}

```

```

\l@yervoffset
64 \newlength{\l@yervoffset}

\m@kelayer \m@kelayer makes a solid layer.
65 \newcommand{\m@kelayer}[1]{%

    The various lengths are calculated. The argument gives the number of the
    layer, i.e. how far down it should be offset from its contents as a multiple of
    \layersize.

66 \setlength{\l@yertotalheight}%
67     {\dp\l@yersavebox+\ht\l@yersavebox+\layersize-#1\layersize}%
68 \setlength{\l@yerlineheight}%
69     {\ht\l@yersavebox-#1\layersize-\fboxrule}%
70 \setlength{\l@yervoffset}%
71     {-\layersize-\dp\l@yersavebox}%
72 \setlength{\l@yerboxwidth}%
73     {\wd\l@yersavebox+\layersize-#1\layersize}%

    The layer is set in a parbox of width \l@yerwidth.
74 \parbox{\l@yerwidth}{%

    Inter-line and inter-paragraph spacing are turned off.

75 \offinterlineskip%
76 \parskip=0pt%

    The contents of the layer is set first.
77 \usebox{\l@yersavebox}%

    The extra “corner” is added on to the bottom right.

78 \rule[\l@yerlineheight]{\layersize}{\fboxrule}%
79 \kern-\fboxrule%
80 \rule[\l@yervoffset]{\fboxrule}{\l@yertotalheight}%
81 \kern-\wd\l@yersavebox\kern-\layersize\kern#1\layersize
82 \rule[\l@yervoffset]{\fboxrule}{\layersize}%
83 \kern-\fboxrule
84 \rule[\l@yervoffset]{\l@yerboxwidth}{\fboxrule}%
85 }}

\l@yer \l@yer draws a layer. The first argument gives the number of the layer, and the
second its contents.

86 \newcommand{\l@yer}[2]{%
87 \sbox{\l@yersavebox}{#2}%
88 \setlength{\l@yerwidth}{\wd\l@yersavebox+\layersize}%
89 \m@kelayer{#1}}

\m@kedashlayer \m@kedashlayer makes a dashed layer. The code is the same as for \m@kelayer
except for the dash commands.

90 \newcommand{\m@kedashlayer}[1]{%

```

```

91 \setlength{\l@yertotalheight}%
92   {\dp\l@yersavebox+\ht\l@yersavebox+\layersize-#1\layersize}%
93 \setlength{\l@yerlineheight}{\ht\l@yersavebox-#1\layersize-\fboxrule}%
94 \setlength{\l@yervoffset}{-\layersize-\dp\l@yersavebox}%
95 \setlength{\l@yerboxwidth}%
96   {\wd\l@yersavebox+\layersize-#1\layersize}%
97 \parbox{\l@yerwidth}{%
98   \offinterlineskip%
99   \parskip=0pt%
100   \usebox{\l@yersavebox}%
101   \raisebox{\l@yerlineheight}{\hd@shrulerule{\layersize}}%
102   \kern-\fboxrule%
103   \raisebox{\l@yervoffset}%
104     {\parbox[b]{\fboxrule}{\vd@shrulerule{\l@yertotalheight}}}%
105   \kern-\wd\l@yersavebox\kern-\layersize\kern#1\layersize
106   \raisebox{\l@yervoffset}%
107     {\parbox[b]{\fboxrule}{\vd@shrulerule{\layersize}}}%
108   \kern-\fboxrule
109   \raisebox{\l@yervoffset}%
110     {\hd@shrulerule{\l@yerboxwidth}}%
111   }}
112 % \end{macrocode}
113 % \end{macro}
114 %
115 % \begin{macro}{\dl@yer}
116 % |\dl@yer| draws a dashed layer, just like |\l@yer| draws a solid
117 % one.
118 %
119 % \begin{macrocode}
120 \newcommand{\dl@yer}[2]
121   {\sbox{\l@yersavebox}{#2}%
122   \setlength{\l@yerwidth}{\wd\l@yersavebox+\layersize}%
123   \m@kedashlayer{#1}}

```

## 2.6 Stacks

Finally, the commands for drawing a stack of layers.

`l@yercount` counts the number of layers drawn by the stack drawing

`l@yercount`

```
124 \newcounter{l@yercount}
```

`\l@yers` `\l@yers` draws a stack of layers; it is parametrized on the command used to draw a layer (third argument). The first argument is number of layers, and the second is the text to set. The layers are drawn in a loop, using `\l@yersavebox` as an accumulator, and the result is typeset.

```
125 \newcommand{\l@yers}[3]
126   {\setcounter{l@yercount}{1}%
127   \sbox{\l@yersavebox}{#2}%

```

```

128   \whiledo{\not\(\value{1@yercount} > #1\)}%
129     {\sbox{\1@yersavebox}%
130      {#3{\value{1@yercount}}{\usebox{\1@yersavebox}}}%
131     \stepcounter{1@yercount}}%
132   \usebox{\1@yersavebox}%
133   }

```

`\lbox` and `\dlbox` are just wrappers for `\l@yers`. They both default to drawing two layers.

`\lbox`

```

134 \newcommand{\lbox}[2][2]{%
135   \l@yers{#1}{#2}{\l@yer}}

```

`\dlbox`

```

136 \newcommand{\dlbox}[2][2]{%
137   \l@yers{#1}{#2}{\dl@yer}}
138 \endpackage

```