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 dashed lines.

`\dashbox` `\dashbox[`*width*`][`*pos*`]{`*text*`}` works like `\framebox`, but the box is drawn with dashed lines.

`\lbox` `\lbox[`*layers*`]{`*text*`}` draws a stack of boxes 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 dashed lines.

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.

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

2.2 Style parameters

Define and give the default values of the style parameters.

\dashlength

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

\dashdash

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

\layersize

```
\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`.

```
\newcommand{\hd@shrule}[1]{%
    \hbox to #1%
        {\vrule height \fboxrule width \dashdash%
        \cleaders\hbox to \dashlength%
            {\hfill\rule{\dashdash}{\fboxrule}\hfill}\hfill%
        \ifthenelse{\lengthtest{#1 > 2\dashdash}}%
            {\vrule height \fboxrule width \dashdash}{}%
        }}
```

\vd@shrule

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

```
\newcommand{\vd@shrule}[1]{%
    \vbox to #1%
        {\hrule height \dashdash width \fboxrule%
        \cleaders\vbox to \dashlength%
            {\vfill\rule{\fboxrule}{\dashdash}\vfill}\vfill%
        \ifthenelse{\lengthtest{#1 > 2\dashdash}}%
            {\hrule height \dashdash width \fboxrule}{}%
        }}
```

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`

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

`\d@shedboxwidth`

```
\newlength{\d@shedboxwidth}
```

`\d@shedboxtotalheight`

```
\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.

```
\newcommand{\m@kedashbox}{%
```

The height plus depth of the box is calculated.

```
    \setlength{\d@shedboxtotalheight}%
        {\dp\d@shedsavebox+\ht\d@shedsavebox+\fboxsep*2+\fboxrule*2}%
```

The box is raised an appropriate amount, and drawn in a `b`-aligned parbox.

```
    \raisebox{-\fboxrule-\fboxsep-\dp\d@shedsavebox}{%
    \parbox[b]{\d@shedboxwidth}{%
```

Inter-line and inter-paragraph skip are disabled.

```
        \offinterlineskip%
        \parskip=0pt%
```

The top line is drawn; the kern makes the left and right sides line up properly.

```
        \hd@shrule{\d@shedboxwidth}%
        \kern-\fboxrule%
        \par%
```

The left-hand side is now drawn, in a parbox of the correct width.

```
        \parbox{\fboxrule}{\vd@shrule{\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.

```
        \kern\fboxsep%
        \parbox{\wd\d@shedsavebox}%
            {\vspace{\fboxsep}\usebox{\d@shedsavebox}\vspace{\fboxsep}}%
        \kern\fboxsep%
```

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

```
        \parbox{\fboxrule}{\vd@shrule{\d@shedboxtotalheight}}%
        \par%
        \kern-\fboxrule%
        \hd@shrule{\d@shedboxwidth}}%
    }}
```

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

```
\newcommand{\dbox}[1]{%
    \sbox{\d@shedsavebox}{#1}%
    \setlength{\d@shedboxwidth}{\wd\d@shedsavebox+\fboxsep*2+\fboxrule*2}%
    \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`.

```
\def\dashbox{\@ifnextchar[\@dashbox\dbox}
\def\@dashbox[#1]{\@ifnextchar[{\@idashbox[#1]}{\@idashbox[#1][c]}}
\long\def\@idashbox[#1][#2]#3%
    {\setlength{\d@shedboxwidth}{#1}%
    \savebox{\d@shedsavebox}[#1-\fboxsep*2-\fboxrule*2][#2]{#3}%
    \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

```
\newsavebox{\l@yersavebox}
```

\l@yerwidth

```
\newlength{\l@yerwidth}
```

\l@yerboxwidth

```
\newlength{\l@yerboxwidth}
```

\l@yertotalheight

```
\newlength{\l@yertotalheight}
```

\l@yerlineheight

```
\newlength{\l@yerlineheight}
```

\l@yervoffset

```
\newlength{\l@yervoffset}
```

\m@kelayer `\m@kelayer` makes a solid layer.

```
\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`.

```
    \setlength{\l@yertotalheight}%
        {\dp\l@yersavebox+\ht\l@yersavebox+\layersize-#1\layersize}%
    \setlength{\l@yerlineheight}%
        {\ht\l@yersavebox-#1\layersize-\fboxrule}%
    \setlength{\l@yervoffset}%
        {-\layersize-\dp\l@yersavebox}%
    \setlength{\l@yerboxwidth}%
        {\wd\l@yersavebox+\layersize-#1\layersize}%
```

The layer is set in a parbox of width `\l@yerwidth`.

```
    \parbox{\l@yerwidth}{%
```

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

```
        \offinterlineskip%
        \parskip=0pt%
```

The contents of the layer is set first.

```
        \usebox{\l@yersavebox}%
```

The extra "corner" is added on to the bottom right.

```
        \rule[\l@yerlineheight]{\layersize}{\fboxrule}%
        \kern-\fboxrule%
        \rule[\l@yervoffset]{\fboxrule}{\l@yertotalheight}%
        \kern-\wd\l@yersavebox\kern-\layersize\kern#1\layersize
        \rule[\l@yervoffset]{\fboxrule}{\layersize}%
        \kern-\fboxrule
        \rule[\l@yervoffset]{\l@yerboxwidth}{\fboxrule}%
    }}
```

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

```
\newcommand{\l@yer}[2]{%
    \sbox{\l@yersavebox}{#2}%
    \setlength{\l@yerwidth}{\wd\l@yersavebox+\layersize}%
    \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.

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

```
    \setlength{\l@yertotalheight}%
        {\dp\l@yersavebox+\ht\l@yersavebox+\layersize-#1\layersize}%
    \setlength{\l@yerlineheight}{\ht\l@yersavebox-#1\layersize-\fboxrule}%
    \setlength{\l@yervoffset}{-\layersize-\dp\l@yersavebox}%
    \setlength{\l@yerboxwidth}%
        {\wd\l@yersavebox+\layersize-#1\layersize}%
    \parbox{\l@yerwidth}{%
        \offinterlineskip%
        \parskip=0pt%
        \usebox{\l@yersavebox}%
        \raisebox{\l@yerlineheight}{\hd@shrule{\layersize}}%
        \kern-\fboxrule%
        \raisebox{\l@yervoffset}%
            {\parbox[b]{\fboxrule}{\vd@shrule{\l@yertotalheight}}}%
        \kern-\wd\l@yersavebox\kern-\layersize\kern#1\layersize
        \raisebox{\l@yervoffset}%
            {\parbox[b]{\fboxrule}{\vd@shrule{\layersize}}}%
        \kern-\fboxrule
        \raisebox{\l@yervoffset}%
            {\hd@shrule{\l@yerboxwidth}}%
    }}
%    \end{macrocoode}
% \end{macro}
%
% \begin{macro}{\dl@yer}
% |\dl@yer| draws a dashed layer, just like |\l@yer| draws a solid
% one.
%
%    \begin{macrocode}
\newcommand{\dl@yer}[2]
    {\sbox{\l@yersavebox}{#2}%
    \setlength{\l@yerwidth}{\wd\l@yersavebox+\layersize}%
    \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`

```
\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 in a loop, using `\l@yersavebox` as an accumulator, and the result is typeset.

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

```
    \whiledo{\not\(\value{l@yercount} > #1\)}%
        {\sbox{\l@yersavebox}%
            {#3{\value{l@yercount}}{\usebox{\l@yersavebox}}}%
        \stepcounter{l@yercount}}%
    \usebox{\l@yersavebox}%
    }
```

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

`\lbox`

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

`\dlbox`

```
\newcommand{\dlbox}[2][2]{%
    \l@yers{#1}{#2}{\dl@yer}}
⟨/package⟩
```