

Glisterings

Peter Wilson

Fame is no plant that grows on mortal soil,
Nor in the glistening foil
Set off to the world, nor in broad rumour
lies:
But lives and spreads aloft by those pure
eyes
And perfect witness of all-judging Jove;
As he pronounces lastly on each deed,
Of so much fame in heaven expect thy
meed.

*Lycidas, Elegy on a Friend drowned in
the Irish Channel, 1637, JOHN MILTON*

The aim of this column is to provide odd hints or small pieces of code that might help in solving a problem or two while hopefully not making things worse through any errors of mine.

Corrections, suggestions, and contributions will always be welcome.

I wuz framed!

Traditional criminal defence

1 Framing

People ask on `comp.text.tex` about putting boxes around some text, or putting a background colour behind some text. An `\fbox` puts a frame round its contents and a `\colorbox`, from the `(x)color` package, makes a coloured background. However, these boxes don't continue across page breaks and on occasion the text for framing will have a page break in the middle. Donald Arseneau's `framed` package [1] is often the solution for this. The package provides four environments as demonstrated below.

The `framed` environment from the `framed` package puts a frame around its contents. Unlike the usual box commands or environments, `framed` continues over page breaks.

The `shaded` environment, also from the `framed` package, puts a colour (or a shade of gray) behind its contents which may continue across page breaks. Before using this environment you have to define a colour named `shadecolor`. In this case I have used the `xcolor` package and

```
\definecolor{shadecolor}{gray}{0.8}
```

FRAMED ILLUSTRATION

Figure 1: Example framed figure

FRAMED ILLUSTRATION & CAPTION

Figure 2: Example framed figure and caption

The `snugshade` environment is similar to `shaded` and puts a colour (or a shade of gray) behind its contents which may continue across page breaks. Unlike the `shaded` environment, though, the colour does not bleed into the margins. This time I set `shadecolor` as

```
\definecolor{shadecolor}{gray}{0.9}
```

to give a look not so arresting as with the setting for the `shaded` example.

The `leftbar` environment, also from the `framed` package, puts a vertical line at the left side of its contents, which may continue across page breaks.

The package can be useful even if you know that there won't be a page break. For instance a `figure` is constrained to be on one page, but you may want to set off some illustrations from the text.

Figure 1 is produced from the code below; only the illustration is framed, not the caption.

```
\begin{figure}
\centering
\begin{framed}\centering
FRAMED ILLUSTRATION
\end{framed}
\vspace*{-0.5\baselineskip}
\caption{Example framed figure}
\label{fig:framef}
\end{figure}
```

Figure 2, where the entire `figure` environment including the caption is framed, is produced from the code below:

```
\begin{figure}
\begin{framed}\centering
FRAMED ILLUSTRATION \& CAPTION
\caption{Example framed figure and caption}
\label{fig:framefcap}
\end{framed}
\end{figure}
```

If a full frame seems a bit of an overkill then using rules might be appropriate. Figure 3 is produced from the following code, and no package is needed. For illustrative purposes I have used very thick rules

 RULED ILLUSTRATION & CAPTION

Figure 3: Ruled figure and caption (2)

SHADED ILLUSTRATION & CAPTION

Figure 4: Example shaded figure and caption

above and below the illustration; normally I would not have them so bold.

```
\begin{figure}
\centering
\rule{\linewidth}{2pt}\
\vspace*{0.5\baselineskip}
RULED ILLUSTRATION \& CAPTION
\vspace{0.5\baselineskip}
\hrule
\caption{Ruled figure and caption (2)}
\label{fig:ruledfcap}
\rule{\linewidth}{2pt}
\end{figure}
```

Depending on its contents, using shading to delineate an illustration may be an option. Figure 4 is produced by the following code, which uses the framed package's shaded environment.

```
\begin{figure}
\begin{shaded}\centering
SHADED ILLUSTRATION \& CAPTION
\caption{Example shaded figure and caption}
\label{fig:shadefcap}
\end{shaded}
\end{figure}
```

The framed package does have some limitations on what can be included within its environments. floats, footnotes, marginpars and header entries will be lost. Further, the package does not work with the multicol package's page breaking, or other code to perform multicolumn balancing. Some restrictions may be lifted in later versions of the package.

Using the package you can create your own new cross-page environments, based on the `MakeFramed` environment defined in the package and specifying a `\FrameCommand` macro which should draw the frame. The `MakeFramed` environment takes one argument which should contain any adjustments to the text width, applied to `\hsize`, and some form of a `restore` command, such as the package's `\FrameRestore` macro or the L^AT_EX internal `\@parboxrestore`, that restores the text attributes to their normal state. The length `\width` is the width of the frame itself. Some examples are given later. But first...

The frame in the `framed` environment is implemented as an `\fbox` via:

```
\providecommand{\FrameCommand}{%
\setlength{\fboxrule}{\FrameRule}%
\setlength{\fboxsep}{\FrameSep}%
\fbox}
```

where `\FrameRule` and `\FrameSep` are lengths defined by the package. By changing these you can change the rule thickness and spacing of the frame. Here is the definition of the `framed` environment itself, which uses the default `\FrameCommand` defined above.

```
\newenvironment{framed}{%
\MakeFramed {\advance\hsize-\width
\FrameRestore}}%
{\endMakeFramed}
```

where `\FrameRestore` restores some text settings, but not as many as have to be done at the end of a minipage.

The other environments are defined similarly. Both `shaded` and `snugshade` use a `\colorbox` as the framing mechanism.

```
\newenvironment{shaded}{%
\def\FrameCommand{\fboxsep=\FrameSep
\colorbox{shadecolor}}%
\MakeFramed {\FrameRestore}}%
{\endMakeFramed}
\newenvironment{snugshade}{%
\def\FrameCommand{%
\colorbox{shadecolor}}%
\MakeFramed {\FrameRestore\@setminipage}}%
{\par\unskip\endMakeFramed}
```

The `leftbar` environment simply uses a vertical rule.

```
\newenvironment{leftbar}{%
\def\FrameCommand{\vrule width 3pt
\hspace{10pt}}%
\MakeFramed {\advance\hsize-\width
\FrameRestore}}%
{\endMakeFramed}
```

Note that in the `framed` and `leftbar` environments the text is narrower than the normal measure, while in the `shade` environments the text width is unaltered and the shading extends into the margins.

Tyger, tyger, burning bright
 In the forests of the night,
 What immortal hand or eye
 Could frame thy fearful symmetry?

Songs of Experience, WILLIAM BLAKE

2 New frames

In some cases it is relatively easy to define your own framing environment based on the `framed` package,

but I have found that some experimentation is often required.

Perhaps you would like to center and frame some text. Here's how.

```
\newenvironment{narrowframe}[1][0.8\hsize]%
  {\MakeFramed{\setlength{\hsize}{#1}
  \FrameRestore}}%
  {\endMakeFramed}
```

This is the `narrowframe` environment where you can adjust the width with the environment's optional length argument. This example is set with the default width.

Or perhaps you would like something a little smoother:

This is the `roundedframe` environment. It requires the `fancybox` package. Perhaps you could use something other than the `\ovalbox` box from that package to give a different frame.

This is the definition I have used for the environment. You can, of course, change the lengths to suit.

```
\newenvironment{roundedframe}{%
  \def\FrameCommand{%
    \cornersize*{20pt}%
    \setlength{\fboxsep}{5pt}%
    \ovalbox}%
  \MakeFramed{\advance\hsize-\width
  \FrameRestore}}
  {\endMakeFramed}
```

Another request that pops up from time to time on `comp.text.tex` is for an environment to show off examples or questions. Robert Nyqvist [2] answered one of these by providing code based on the `framed` package, that was basically as follows. An example of the `ruledexample` environment is shown below as Ruled Example 1.

```
\makeatletter
\definecolor{rulecolor}{gray}{0.65}
\newcounter{ruledexample}
\newlength{\releftgap}
  \setlength{\releftgap}{4pt}
\newlength{\rerightgap}
  \setlength{\rerightgap}{1em}
\newlength{\rerule}
  \setlength{\rerule}{1.25pt}
\newlength{\Eheight}
\newenvironment{ruledexample}[1][Example]{%
  \settoheight{\Eheight}{\textbf{#1}}%
  \addtolength{\Eheight}{-\rerule}%
  \def\FrameCommand{\hspace{-\releftgap}%
  {\color{rulecolor}%
  \vrule width \rerule}%
  \hspace{\releftgap}\hspace{-\rerule}}%
  \MakeFramed{\advance\hsize-\width}%
  \refstepcounter{ruledexample}%
  \makebox[0pt][1]{%
  \hspace{-\parindent}%
  \hspace{\rerightgap}%
  {\color{rulecolor}\rule{1.5em}{\rerule}}%
  \quad
  \raisebox{-0.5\Eheight}[0pt]{%
  \textbf{#1} \theruledexample}}%
  }[[.5\baselineskip]%
  \noindent\ignorespaces}%
  {\@afterheading\
  \makebox[0pt][1]{%
  \hspace{-\releftgap}%
  {\color{rulecolor}
  \rule{\columnwidth}{\rerule}%
  \rule{\releftgap}{\rerule}}%
  }}
  \endMakeFramed}
\makeatother
```

```
\hspace{\releftgap}\hspace{-\rerule}}%
\MakeFramed{\advance\hsize-\width}%
\refstepcounter{ruledexample}%
\makebox[0pt][1]{%
  \hspace{-\parindent}%
  \hspace{\rerightgap}%
  {\color{rulecolor}\rule{1.5em}{\rerule}}%
  \quad
  \raisebox{-0.5\Eheight}[0pt]{%
  \textbf{#1} \theruledexample}}%
}[[.5\baselineskip]%
\noindent\ignorespaces}%
{\@afterheading\
\makebox[0pt][1]{%
  \hspace{-\releftgap}%
  {\color{rulecolor}
  \rule{\columnwidth}{\rerule}%
  \rule{\releftgap}{\rerule}}%
  }}
\endMakeFramed}
\makeatother
```

— Ruled Example 1

This is the `ruledexample` environment, which is titled and numbered and can be `\labelled`, and which will break across pages if need be. The code basis was originally posted to `comp.text.tex` by Robert Nyqvist in 2003 but I have added some extra means of controlling the rules and spacing.

- `rulecolor` is the color of the rules
- `\rerule` is the thickness of the rules
- `\releftgap` is the distance the vertical rule is moved into the margin
- `\rerightgap` is the indentation of the title rule
- You can use the optional argument to specify the title, which by default is 'Example'.

As you can see by this example, you can use code like `itemize` within a framed environment.

Some users have found that the behaviour of the `framed` environment within a list environment such as `quote`, or `adjustwidth` from the `changepage` package [4], is not quite what they expected. This is described in a little more detail below

This is the start of a quotation environment. You can use the `framed` environment within this, and also any other `list` based environment.

We are now in the `framed` environment, and notice how the frame extends across the original textwidth. This is not always what you might want.

Donald Arseneau and I had an exchange of views, and code, on this and we each produced code that created a closer frame. As usual Donald's code was much better than mine.

Now we are in the `qframe` environment based on Donald Arseneau's code and which I included in my memoir class [3]. The class also provides a 'shaded' version.

If you happen to use the `adjustwidth` environment, from the `changepage` package¹ [4] or the memoir class, then the `qframe` environment works better in it than does the `framed` environment.

Here endeth the `quotation` environment.

This is the definition of the `qframe` environment:

```
\makeatletter
\newenvironment{qframe}{%
  \def\FrameCommand##1{%
    \setlength{\fboxrule}{\FrameRule}%
    \setlength{\fboxsep}{\FrameSep}%
    \hskip\@totalleftmargin\fbox{##1}%
    \hskip-\linewidth
    \hskip-\@totalleftmargin
    \hskip\columnwidth}%
  \MakeFramed{\advance\hsize-\width
    \advance\hsize \FrameSep
    \@totalleftmargin\z@
    \linewidth=\hsize}}%
  {\endMakeFramed}
\makeatother
```

If you put this code in a class (`.cls`) or package (`.sty`) file then you do not need the `\makeatletter` and `\makeatother` pairing, otherwise you do.

Another potentially useful sort of frame is where there is a title—one that gets repeated after a page break. The code for this is considerably more complicated than earlier, and I had to do quite a lot of experimentation to get it to work. First, though, an example.

A 'framewithtitle'

Using the `framewithtitle` environment, which this is, you can put a title inside the frame. If the contents extend past a pagebreak then a 'continued' title is put at the start of the frame on the succeeding pages. The code for `framewithtitle` is given in this article.

The position of the title is set by the `\titleframe` command which is initially defined

¹ `changepage` is the successor to the `chnpage` package which is being phased out.

A 'framewithtitle' (cont.)

as below. If you would prefer the title to be centered, then change it like:

```
\newcommand*{\frametitle}[1]{%
  \strut#1}%          at left
\renewcommand*{\frametitle}[1]{%
  \centerline{\strut#1}}%      centered
```

This example shows that you can include verbatim text in a framed environment.

Here is the code for the `framewithtitle` environment. It is based on hints from Donald Arseneau that applied to an earlier version of `framed`. A lot of the code is concerned with handling the title so that the first can be used initially, then after a page break a continuation title will be used instead. I have also extracted a lot of code from my original version as separate macros as I'm going to reuse much of it later. My most difficult problem was that internally the package keeps resetting the text to determine where any page breaking will occur, and only after that does it actually typeset; I couldn't assume that the initial title would be immediately set and had to reserve its typesetting until after the all the internal resetting had been completed; I used `\ifcontframe` for this purpose. The macro `\FrameSetup` does most of the work related to the titling while `\FrameTitle` actually typesets the title(s). The required argument to the `framewithtitle` environment is the initial title, and you can use the optional argument to override the default continuation title.

```
\makeatletter
\newcommand*{\frametitle}[1]{\strut#1}
\newif\ifcontframe

\newcommand*{\FrameSetup}[2]{%
  \fboxrule=\FrameRule \fboxsep=\FrameSep
  \global\contframefalse
  \def\FrameFirst{\textbf{#2}}%
  \def\FrameCont{\textbf{#1}}%
  \def\FrameCommand##1{%
    \Title@Frame{\FrameCurrent}{##1}%
    \global\let\FrameCurrent\FrameNext
  \ifcontframe
    \global\let\FrameNext\FrameCont
  \fi
  \global\contframetrue}%
  \global\let\FrameCurrent\FrameFirst
  \global\let\FrameNext\FrameFirst}

\newcommand*{\FrameTitle}[1]{%
  \nobreak \vskip -0.7\FrameSep
  \rlap{\frametitle{#1}}%
  \nobreak\nointerlineskip
  \vskip 0.7\FrameSep}
```

```

\newenvironment{framewithtitle}[2]%
  [\FrameFirst\ (cont.)]{%
  \def\Title@Frame##1##2{%
    \fbox{\vbox{\FrameTitle{##1}%
    \hbox{##2}}}}%
  \FrameSetup{#1}{#2}%
  \MakeFramed{%
    \advance\hsize-\width
    \FrameRestore}}%
  {\global\contframefalse
  \endMakeFramed}
\makeatother

```

As an alternative to `framewithtitle` you can use the `titleframed` environment where the title is set outside the frame. The arguments to the two environments are the same, as is setting the title's horizontal position.

A 'titleframed'

With the `titleframed` environment, which this is, you can put a title on top of the frame. If the contents extend past a pagebreak then a 'continued' title is put at the start of the frame on the succeeding pages. The code for `titleframed` is given in this article.

The position of the title is set by the `\titleframe` command which is initially defined as below. If you would prefer the title to be centered, then change it like:

```

\newcommand*\frametitle[1]{%
  \strut#1}% at left
\renewcommand*\frametitle[1]{%
  \centerline{\strut#1}}% centered

```

which is what I have done for this example.

So, here is the code for `titleframed` which, as you can see, shares most of the code with its companion environment.

```

\makeatletter
\newenvironment{titleframed}[2]%
  [\FrameFirst\ (cont.)]{%
  \def\Title@Frame##1##2{%
    \vbox{\FrameTitle{##1}%
    \noindent\fbox{##2}}
  \FrameSetup{#1}{#2}%
  \MakeFramed{%
    \advance\hsize-\width
    \advance\hsize -2\FrameRule
    \advance\hsize -2\FrameSep
    \FrameRestore}}%
  {\global\contframefalse
  \endMakeFramed}
\makeatother

```

I admit that I have given little explanation of the code examples. The excuse that I would like you to believe is that adding all the explanatory material would make the article too long, but the real reason is that I do not really understand how the `framed` package performs its magic; hence the experiments (i.e., many trials and even more errors) that I did to create code that seemed workable.

References

- [1] Donald Arseneau. The framed package v0.95, 2007. mirror.ctan.org/macros/latex/contrib/framed.
- [2] Robert Nyqvist. 'example' environment or command. Post to `comp.text.tex` newsgroup, 11 January 2003.
- [3] Peter Wilson. The memoir class for configurable typesetting, 2008. mirror.ctan.org/macros/latex/contrib/memoir or mirror.ctan.org/install/macros/latex/contrib/memoir.tds.zip.
- [4] Peter Wilson. The changepage package, October 2009. mirror.ctan.org/macros/latex/contrib/changepage.

◇ Peter Wilson
 12 Sovereign Close
 Kenilworth CV8 1SQ, UK
 herries dot press (at)
 earthlink dot net