

Hints & Tricks

Glisterings

Peter Wilson

Remember still that the loftier minde
That in this world doth seek to glister so,
Blowne on this rock by fonde vainglorious
winde,
Falls headlong down to everlasting wo.

The Ship of safegarde, (1569)
BARNABE GOOGE

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.

Addresses are given to us to conceal our whereabouts.

Reginald in Russia,
SAKI

1 Address lists

For many a year I have been promising my wife that I would print labels for the envelopes for our Christmas cards. I even went as far as buying some software to run on an OS that went out of date in the last century. This year (2005) I have at last salved this part of my conscience with the aid of L^AT_EX and Boris Veytsman's EnvLab package [1].

After some experiments I created a package file, myenvlab, that gave me the setup that I wanted. I found that I had to use the EnvLab's \SetLabel command to adjust the address spacing to match the sheets of labels that I was planning to use. I also organised things so that each label could be framed by an \fbox and I could then print a page of addresses onto an ordinary sheet of paper to check if the spacing was correct for the real label sheets. For further information consult the package documentation.

```
% file myenvlab.sty
\usepackage[avery5160label,
             noprintbarcodes,
             nocapaddress]{envlab}

%% Subtract 0.1 inch from vertical dimensions!!
\SetLabel{4.19in}{1.23in}{0.73in}{0.16in}%
           {0.19in}{2}{7}

\newif\ifboxlabel
\let\oldPrintLabel\PrintLabel
\renewcommand{\PrintLabel}[1]{%
  \ifboxlabel
```

```
    \fbox{\oldPrintLabel{#1}}%
\else
  \oldPrintLabel{#1}%
\fi}
\boxlabeltrue
\endinput
```

The myenvlab package could then be used in a file like the one below to print out a set of address labels, where the \add macro defined the name and address for a label.

```
% file xmas.tex
\documentclass[12pt]{letter}
\usepackage{myenvlab}
%%\boxlabelfalse
\newcommand{\add}[1]{%
  \mbox{\mlabel{\mbox{}}{#1}}
\newcommand{\UK}{UNITED KINGDOM}
\startlabels
\begin{document}
\add{John Doe \
      98765 931st St \
      \ Someplace YN 12345}
\add{A N Other \
      The House \
      \ The Road \
      \ Town \
      \ \UK}
% etc., etc.
\end{document}
```

Having printed out sheets of labels it occurred to me that probably other labels would be required at other times. My wife was also talking about starting a new address book because the current one was becoming illegible due to many deletions, changes, and additions. I was toying with the idea of subverting BIB_TE_X into an address database but fortunately hesitated before having the joy of programming with the BIB_TE_X language. I now keep all the addresses in a file that looks like this, where I specify a macro for each name and address, and any other personal details that might be of interest:

```
% file addresslist.tex
%%\ \add{name}{address}{telephone}{email}{notes}
\newcommand*{\UK}{UNITED KINGDOM}
\newcommand*{\DoeJ}{\add{John Doe}%
  {98765 931st St \
  \ Someplace YN 12345}%
  {(981) 123--4567}%
  {\url{jd576@email.moc}}%
  {birthday 01/01/01}}
\newcommand*{\OtherAN}{\add{A N Other}%
  {The House \
  \ The Road \
  \ Town \
  \ \UK}%
  {+44 1273 5798 8975}%
  {\url{ano@ano.org}}%
  {dog: Fido}}
% etc., etc.
\endinput
```

Using a suitable definition for \add, which in this case puts the various arguments into a mini-page, I can print an address book by:

```
% file addressbook.tex
\documentclass[12pt,twocolumn]{article}
```

```
\usepackage{url}
\newcommand{\add}[5]{%
  \begin{minipage}{\linewidth}\raggedright
    #1 \ \ #2 \ \ #3 \ \ #4 \ \ #5 \end{minipage}%
  \\[\baselineskip]}
\begin{document}
\input{addresslist}
\DoEJ
\OtherAN
% etc., etc.
\end{document}
```

With a different definition for `\add`, which here just uses the first two arguments, namely the name and address, I can print labels by:

```
% file xmas.tex
\documentclass[12pt]{letter}
\usepackage{url}
\usepackage{myenvlab}
\newcommand{\add}[5]{%
  \mbox{} \mlabel{\mbox{}}{#1\#2}}
\startlabels
\begin{document}
\input{addresslist}
\DoEJ
\OtherAN
% etc., etc.
\end{document}
```

Now, *here*, you see, it takes all the running *you* can do, to keep in the same place. If you want to go somewhere else, you must run at least twice as fast as that.

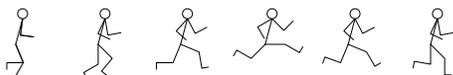
Through the Looking Glass,
LEWIS CARROLL

2 Animated books

While sorting through some old files I found a piece written by Jeremy Gibbons for his Hey — it works! column but which was not published before he handed his baton over to me. Jeremy kindly gave me permission to include it here.

When I was a child, my father used to make little booklets, each page with a slightly different picture; flicking through the booklet quickly makes a ‘movie’. Recently James Willans from York asked on `comp.text.tex` how to achieve this effect in L^AT_EX, and a Michael Liebling answered. Here I show a simpler version of Liebling’s approach.

First you need a collection of little pictures; the following assumes that they are all the same size. I used METAPOST to generate a running man in different positions. Here is a representative sample:



The METAPOST file `running.mp` to create these figures is linked from the TUGboat web page <http://tug.org/TUGboat/Contents/contents27-1.html>.

Next we compute the page number modulo the number of different pictures:

```
\def\compute@modulus#1#2{%
  \@tempcnta=#2\relax
  \divide\@tempcnta by #1\relax
  \multiply\@tempcnta by #1\relax
  \multiply\@tempcnta by -1\relax
  \advance\@tempcnta by #2\relax}
```

So for example `\compute@modulus{12}{\thepage}` computes the page number modulo twelve.

We also work out how far to move the image on each page, dividing the difference between the text width and the image width by the number of pages:

```
\newcount\pagecount
\pagecount=100
\setbox0=\hbox{%
  \includegraphics[scale=0.5]{running.0}}
\newdimen\distance
\distance\textwidth
\advance\distance by -\wd0
\divide\distance by \pagecount
```

Finally, we use the `plain` page style, and put the right image in the right place on each page:

```
\newdimen\offset
\def\@oddfoot{%
  \offset=\distance
  \multiply\offset by \thepage
  \hskip\offset
  \compute@modulus{12}{\thepage}%
  \includegraphics[scale=0.5]{
    {running.\the\@tempcnta}%
  \hfil}
\let\@evenfoot\@oddfoot
```

All that remains is to generate the right number of pages:

```
\loop
  \mbox{} \newpage
\ifnum \pagecount>0
  \advance\pagecount by -1
\repeat
```

References

- [1] Boris Veytsman. Printing Envelopes and Labels in L^AT_EX 2_ε: EnvLab Package User Guide, June 1996. Available on CTAN in `latex/macros/contrib/envlab`.

◇ Peter Wilson
18912 8th Ave. SW
Normandy Park, WA 98166
USA
herries dot press (at) earthlink dot
net