‘Hey — it works!’

Jeremy Gibbons

Welcome once again to Hey — it Works!, a home for intriguing, elegant or just plain useful (L)ATeX techniques. Please send me any ideas that you have for articles. Back issues of the column are available from my web page.

This issue we have four articles. The first, by my ex-colleague Paul Hafner, explains how to include in a document the page range of the document. The second is my own, and shows how to get text italics instead of math italics in maths mode by default. We conclude with two short pieces: one by Ramón Casares, showing how to fill the last line of each paragraph with a rule to prevent forgery by the addition of extra text, and one by Donald Arseneau, showing how to produce a ‘closed surface integral’ symbol.

1 Determining the page range of a document

It is customary for journals to have an imprint on the first page of each article identifying the journal and providing the bibliographic details (volume, year, page range). LATEX can provide the page information, but it does not come without an effort. The page number of the first page is easily found: place a label at the start of the article, most conveniently using the command

\AtBeginDocument{\label{firstpage}}

Then the page number of the first page will be available via \pageref{firstpage}.

For the last page one is tempted to opt for the symmetric solution:

\AtEndDocument{\label{lastpage}}

However, the document LATEX 2ε for class and package writers (file clsguide.tex) explains carefully that the label will be placed before any end floats. The remedy provided in clsguide is to insert \clearpage before the label:

\AtEndDocument{\clearpage\label{lastpage}}
However, if you follow this all too literal reading of
the documentation, the label will remain undefined,
i.e., nothing will be written to the .aux file.

Now a solution is obvious: instead of \label we
insert the code which writes the appropriate infor­mation to the .aux file (we steal it from latex.ltx).
Since the \clearpage has moved us onto the ‘page’
after the last, we decrement the page counter before
writing it out, and (for completeness) increment it
again afterwards.

\AtEndDocument{\clearpage
\addtocounter{page}{-1}%\immediate\write\@auxout{%\string\newlabel{lastpage}%{{}{\thepage}}}%\addtocounter{page}{1}%}

In fact, at the CTAN archives there is a package
\texttt{lastpage} by Jeff Goldberg that does exactly this.

An alternative implementation of the same idea
redefines the command \texttt{\enddocument} (defined
in the file latex.ltx), inserting the additional code
after the \texttt{\clearpage} in line 4 of this definition.
This has the advantage that collisions with other
insertions via \texttt{\AtEndDocument} (say, if the article uses
other packages) are less likely (such insertions might
assume that the \texttt{\clearpage} command has not yet
been issued; in this case we have to be very careful
about the order in which they are loaded). On the
other hand, to remain fully compatible and up to
date, the maintainer of a package which redefines a
critical command such as \texttt{\enddocument} has a
six-monthly maintenance task for life: to check if
a new release of \TeX has introduced changes of
this command, and to accommodate them. I cannot
foresee too many occasions where one would like to
insert code at that late stage (lack of imagination?)
but there could be a case for a \texttt{very-end-hook} at the
position in question, or maybe a routine label for
the last page that can be accessed by everyone.

\begin{itemize}
\item Paul Hafner
  Department of Mathematics,
    University of Auckland
  hafner@math.auckland.ac.nz
\end{itemize}

2 Text italics in maths mode

In most of mathematics, it is conventional to use
single letters for variables. If two letters are adja­
cent, as in ‘zy’, this is read as two separate quan­tities
\(x\) and \(y\) multiplied together. For this reason,
\TeX and \LaTeX use two italic fonts: ‘text italics’ for
italic characters in text, and ‘math italics’ for
italic characters in maths. The difference is clearly
visible with regard to ligatures (‘\textit{difficult’} versus
‘\textit{difficult’}), but most of the letters have slightly
different shapes in the two fonts.

In contrast, in my corner of computer science,
it is common to use words instead of single let­ters as variable names; for example, I might write
‘flatten \texttt{xss}’. This looks horrible using the default
math italics. I can get round the problem by using
\texttt{\mathit} to select text italics for every variable men­
tioned in my document; this looks fine (‘flatten \texttt{xss}’)
but is a pain to type. It would be nice to make letters
appear in text italics by default.

It is tempting to use \texttt{\SetSymbolFont} for this:

\begin{verbatim}
\SetSymbolFont{letters}{normal} {OT1}{cmr}{m}{it}
\end{verbatim}

This makes maths family 1, from which letters in
maths mode are normally taken, use text italics
instead of math italics. This does achieve the desired
effect, but it has some unfortunate side-effects too.
Not only the letters, but also many other symbols
are taken from family 1. Because Computer Modern
text italics has a different encoding from math
itals, many of these other symbols now appear as
the wrong glyph. For example, full stops appear as
colons, commas as semi-colons, the Greek letters
are all wrong, and so on. Of course, the definitions
of all these symbols can be changed, to take them
from other families, but this is a both nuisance
and error-prone: a quick count through \texttt{fontmath.ltx}
reveals that more than 50 definitions need changing.

Fortunately, there is a better solution. This is
to declare a new symbol font for text italics:

\begin{verbatim}
\DeclareSymbolFont{textit}{OT1}{cmr}{m}{it}
\end{verbatim}

and to take each of the letters from this new symbol
font:

\begin{verbatim}
\DeclareMathSymbol{0}{\mathalpha}{textit}{'0}
\end{verbatim}

This gives, for example, ‘\texttt{inits1}’ instead of ‘\texttt{inits1}’.
However, this makes all digits in maths mode italic,
which may not be what you want; somehow, the
number \(3.14159265\) does not look quite right.

\begin{itemize}
\item Jeremy Gibbons
  Oxford Brookes University
  jgibbons@brookes.ac.uk
\end{itemize}
3 Anti-appending rule

When important documents, as contracts, were hand written it was usual, at least in Spain, to complete the last line of every paragraph with a line. The purpose of the line was to avoid the completion of the paragraph after signing the document to change the meaning of the agreement.

It is in fact an amusing exercise to take the last document written by, let’s say, the boss and make a nonsense of it just using the part of the line that was left blank in each paragraph.

If you fear that one of your documents could be faked by your enemies or if you just want to imitate the old legal style of writing, then \TeX makes it easy.

\def\unfakeablepar\unskip\nobreak
 \leaders\parrule\hskip\parfillskip
 \vadjust{\{}\parfillskip=0pt\endgraf}}
\def\parrule{\hrule height 2.2pt depth -1.8pt\relax}

To activate the feature just write
\let\par=\unfakeablepar

To return to the fakeable style write
\let\par=\endgraf

\begin{itemize}
  \item Ramón Casares
    Telefónica de España
    r.casares@computer.org
\end{itemize}

4 Closed surface integral

Donald Arseneau sent in this quick definition of a closed surface integral $\ointint$, a double integral like $\oint$:

$\ointint$

This is defined by:
\begin{verbatim}
\newcommand\ointint{{\begingroup \displaystyle \unitlength 1pt
  \int\mkern-7.2mu\begin{picture}(0,8)
  \put(0,8){\oval(10,8)}
  \end{picture}\mkern-7mu\int\endgroup}
\end{verbatim}

Note that if we had used braces {...} here instead of \begingroup...\endgroup, the subscript would be attached to the whole \ointint subformula (as on the left, below) instead of being attached to the second \int and tucking neatly under it (as on the right):

$\ointint_S \quad \ointint_S$

\begin{itemize}
  \item Donald Arseneau
    Tri-University Meson Facility
    asnd@reg.triumf.ca
\end{itemize}