

time required to do the necessary work, the idea was largely ignored. Also there was the problem of obtaining appropriate versions of the manuscripts, some of which were still undergoing editing revisions, so were not yet finalised.

For some time I've been routinely preparing mathematical course materials for paper (via \LaTeX) and in HTML, using $\LaTeX2HTML$. Recently I started using *pdfTeX* as well, and organise the manuscripts to process seamlessly with all three tools, while exploiting the best features of each. Having some time available, working on a proceedings collection in PDF seemed like an appropriate thing to do, and could provide valuable experience for similar work in the future.

There are two main tasks here:

- Prepare a PDF version of each paper.
- Somehow combine the papers, using active hyperlinks, to present as if part of a unifying electronic document.

As each task separately requires some amount of editing within each author's manuscript, I wanted to develop a method which would minimize the number of times each file need be manually edited. It was felt that any decisions concerning styles and layout should be able to be applied to all the preprints, without any need to make edits in the individual files. To a large extent this was achieved. The results of this work can be viewed at <http://www.tug.org/TUG99-web/program.pdf> which has links to .pdf files for the papers, in the directory <http://www.tug.org/TUG99-web/pdf/>.

Below I describe the techniques developed, and lessons learned. Some of these lessons and techniques are doubtless known already to experienced \LaTeX and \TeX users; others are new and can surely be refined to become even more useful. I'm writing this article with hindsight², after the TUG'99 meeting has concluded; indeed some of the work on individual preprints was done on returning home after the meeting. Advice is given, to help authors simplify the tasks of editors, which in turn leads to reducing the time required for a publication to be prepared.

Preparation Notes

The main issues for creating PDF, as distinct from DVI, versions of papers submitted for a proceedings (or any other) volume relate to

- A. bookmarks—navigation to sections, subsections, figures, etc.;

Report

Preparation of Documents for Multiple Modes of Delivery—Notes from TUG'99

Ross Moore

Background

As the theme of the TUG'99 meeting concerned preparing documents for Web-based delivery, and the \TeX -related tools recently developed for this purpose, it had been suggested¹ that it would be nice to apply some of these tools to the preprints for the meeting, in order to show off the effectiveness of these tools. As no single person had the expertise in all of *pdfTeX*, $\LaTeX2HTML$ and $\TeX4ht$, nor the

¹ by Mimi Burbank

² ... and at the request of Christina Thiele

- B. active internal hyperlinks for citations and cross-references;
 - C. active hyperlinks to external URLs mentioned in the paper, and/or other papers in the same proceedings set;
 - D. incorporation of included graphics.
- Of course, also of relevance is:
- E. how the papers, as individual .pdf files, will be linked back to a common document which serves as a wrapper, including a Table-of-Contents with active hyperlinks to each paper.

Since most of the submissions for TUG'99 were prepared using L^AT_EX, and the `hyperref` package already provides an automatic solution to issues A and B (provided the author has used `\label`, `\ref` and `\cite` appropriately) it was decided to use *pdf*T_EX, via the `pdflatex` command, and `hyperref` for *all* the papers. This includes the papers originally submitted as T_EX source, rather than L^AT_EX, for which there would necessarily be some extra editing required. Thanks to Sebastian Rahtz and other authors, the packages and drivers to tackle issues C and D were already available, so it was not necessary to write any complicated macros to implement these effectively.

To obtain a consistent style across all the papers, and to ensure that the same packages are available for handling citations, URLs, graphics etc. it was decided to use a common “driver” file, implemented as follows.

- Each submitted paper was stored in a separate subdirectory, along with any styles, graphics and bibliography files. (This structure was already in place, due to earlier phases of the editing process.)
- A common file, called `TUG99pdf.pre` was located in the common parent of these subdirectories. This file would be `\input` at the beginning of each job. This file contains the `\documentclass` command, and commands to load suitable packages. Parts of its contents will be described in due course.
- For each paper a “mini-driver” file was made, to load `TUG99pdf.pre` and subsequently `\input` the author’s original source (or rather, the current version available in the editing process). This file was named e.g. `rowley.ltx`, where the current source revision is `rowley5.ltx`. It is this mini-driver file that is actually typeset, to produce `rowley.pdf` and auxiliary files.

For example, the mini-driver for most of the L^AT_EX submissions was as follows:

```
\input ../TUG99pdf.pre
\input{\jobname\revisionLevel.ltx}
```

Notice that the name of the paper to be processed does not occur explicitly within this file. It is constructed from `\jobname` and `\revisionLevel` (set to 5 within `TUG99pdf.pre`). Thus it is sufficient to have a single file `tug99art.ltx` within the parent directory. Then `rowley.ltx` is just a symbolic link to `../tug99art.ltx`.

For those authors that chose to use the Harvard style of citation, there is a similar mini-driver, called `tug99harv.ltx`, with contents:

```
\PassOptionsToClass{harvardcite}{ltugproc}
\input ../TUG99pdf.pre
\input{\jobname\revisionLevel.ltx}
```

Notice the use of `\PassOptionsToClass`, to ensure that appropriate code is used when `\documentclass` is subsequently encountered.

To prevent `\documentclass` being run twice in the same job, the file `TUG99pdf.pre` concludes with:

```
\renewcommand{\documentclass}[2] [] {}
\let\usepackage\RequirePackage
\let\newcommand\providecommand
```

This way packages loaded from within the author’s source do not cause conflicts (e.g. with options or drivers) when already loaded from `TUG99pdf.pre` or from `ltugproc.cls`. Similarly by forcing the use of `\providecommand`, instead of `\newcommand`, within the author’s manuscript, name-clashes are avoided when the author tries to define a command-name that is already available. Indeed the author’s attempt is ignored completely, so that a consistent style is maintained across all the submitted papers. For example, `\DVI` is defined by `ltugproc.cls` to expand to `\acro{dvi}`, however an author may try to define `\newcommand{\DVI}{\texttt{dvi}}`. Using `\providecommand`, the author’s attempt is ignored, so that any adjustments to the expansion of `\acro` will be applied in this paper also.

Advice to Authors: *Get into the habit of using `\providecommand` for stylistic markup, whenever it is conceivable that your document may become part of a journal issue or edited volume. Reserve use of `\newcommand` for text-replacements or macros that are guaranteed to be specific to your own manuscript.*

Similarly, use `\RequirePackage` whenever possible, rather than `\usepackage`, as this allows easier integration of your source with packages and styles for the journal or edited volume.

This use of a driver-file has effectively implemented Kaveh Bazargan's idea³ of using two different class files. An author uses `ltugproc.cls` while preparing his/her manuscript, while the editors use whatever class is requested from `TUG99pdf.pre`. For the record, `TUG99pdf.pre` starts as follows:

```
\PassOptionsToPackage{pdftex,colorlinks,
  linkcolor=blue,citecolor=magenta}{hyperref}
\documentclass{ltugproc}
\RequirePackage{latin1}{inputenc}
\RequirePackage{url}
\RequirePackage{html}
\RequirePackage{graphicx}
\RequirePackage{enumerate}
\RequirePackage{alltt}
...
```

Notice that the `hyperref` package is not explicitly requested, since it will be loaded automatically from the `html` package, available with the most recent revisions of `LATEX2HTML`, when the processing is being done by `pdfTEX`.

Bookmarks. Automatic bookmarks are created for section and subsections, and also (optionally) for figures and tables, which provides a useful alternative to a Table of Contents, and List of Figures, etc. However, only plain text is allowed for the text of the active hyperlink in the bookmark window. This means that section headings cannot contain styled text, or mathematics, unless an alternative simplified optional argument is supplied. Similarly an optional argument should be provided for complicated, or long, figure captions.

Advice to Authors: *Get into the habit of providing optional arguments to section titles and figure/table captions, if only as a comment to be used if required. For example:*

```
\section
%[pdfTeX and LaTeX] % uncomment if needed
{\pdfTeX{} and \LaTeX}
...
```

Internal Hyperlinks. `LATEX`'s `\label` and `\ref` mechanism translates directly into active hyperlinks in the PDF document when the `hyperref` package has been loaded. Similarly `\cite` commands produce active links to the bibliography listing, at least with some of the available packages for formatting citations and bibliographies. Patrick Daly's `natbib` package is generally the best to use, and is fully supported by `hyperref` for `pdfTEX` (and `LATEX2HTML`). The Harvard style of citation is also supported by `natbib` by loading it with an optional argument:

```
\usepackage[nharvard]{natbib}
```

so there is no excuse for the die-hards not to use it.

Advice to Authors: *Learn to use L^AT_EX's symbolic \label-\ref mechanism, if you don't already do so. With electronic documents processed by either pdfT_EX or L^AT_EX2HTML, the cross-references will become active hyperlinks, which are far more useful than a number or other passive marker. Similarly learn to use natbib for the bibliography and citations.*

External Hyperlinks. The best package for formatting URLs is undoubtedly Donald Arseneau's `url.sty`, which can be used with either `LATEX` or `TEX`. It is supported by both `hyperref` and `LATEX2HTML`, to create active hyperlinks in PDF and HTML documents respectively.

A common practice among authors is to typeset URLs using `\texttt` or `{\tt ...}`. This is *visual* markup, not *logical* markup, and should be avoided within the body of the document. It is better to use a `LATEX`-like notation: `\myurl{...}` even if the definition is just `\def\myurl#1{\tt #1}`. This allows an editor to load `url.sty` and insert a single line: `\let\myurl\url` into the preamble of the document to make the hyperlinks active.

There are two quite common errors with URLs. Firstly, don't forget the `http://` at the start, or `ftp://`, or whatever else is appropriate. Acrobat Reader, or a Web-browser, interprets `www.tug.org` as a *relative* URL, resulting in an error.

If a relative reference is indeed intended, e.g. to a directory relative to the author's home-page, then make sure that a valid URL to the home-page is provided within the document preamble. The syntax used by `hyperref` for this is

```
\hyperbaseurl{http://www.tug.org}
```

Even if your document doesn't use `hyperref`, it is useful to include such a line, commented-out, where it can be easily found by the journal editor.

The second common pitfall is in using a notation such as: `CTAN/macros/latex/supported`. While any *TUGboat* reader will understand exactly what is meant, the resulting hyperlink will fail in a browser, since it will be assumed to be a relative URL. If you really wish that string to be displayed, mark it up as:

```
\texttt{CTAN/}\url{macros/latex/supported}
```

and provide a valid `\hyperbaseURL`, such as:

```
ftp://ctan.tug.org/ctan
```

Advice to Authors: *Read and understand the issues discussed in the preceding paragraphs.*

³ in this volume

Another type of active hyperlink can be very effective in an electronic document. For example, every mention of “Adobe Acrobat” or perhaps just the first, can be a hyperlink to the download page to obtain the latest version of the software. Such links are especially useful in bibliography listings, where they can provide a direct link to an electronic version of a cited paper, or to a preprint archive, or a publisher’s Web site. Commands for this are `\href` from `hyperref` and `\htmladdnormallink` from `html.sty`.

Included Graphics. Using `pdfTeX` to create the PDF files, it is not possible to include PostScript graphics directly. Instead they must first be converted to PDF, then these can be included as part of the job. The conversion can be done using either Ghostscript, or with Acrobat Distiller. A *Perl* script `epstopdf`, by Sebastian Rahtz and Thomas Esser, creates the correct command for Ghostscript, after having first read the `%%BoundingBox` comment to establish the correct size for the translated image. Alternatively the script `ps2pdf` uses Ghostscript to convert full pages to full PDF pages; if this is more than what is required, it should still be possible to crop the image when it is included in the PDF document. For PostScript files which are not EPS, or for which there is no `%%BoundingBox` comment, then Ghostscript can create a valid EPS version, prior to using `epstopdf`.

As for including the image within the document, the best \LaTeX command to use is the version of `\includegraphics` from the `graphicx` package. Its optional argument is flexible enough to be able to do anything that is possible with other commands, such as `\psfig` or `\epsfbox`. Furthermore, with `\includegraphics` it is not necessary to include the `.eps` suffix with the filename, since this is the default when a graphics file of this type exists. Similarly when `pdfTeX` is used, the default is `.pdf`, or `.jpg` when there is no appropriate `.pdf` file in the search paths. Hence the codeline

```
\includegraphics[scale=.5]{myimage}
```

suffices to include the correct version of the graphic, either `myimage.eps` with DVI, or `myimage.pdf` or `myimage.jpg` with the PDF version.

Advice to Authors: *Check all Encapsulated PostScript graphics for correct `%%BoundingBox` information. Load the `graphicx` package and become acquainted with the possibilities available with the optional argument to `\includegraphics`. Also look at the `\DeclareGraphicsRule` command, if `.jpg` or other graphic formats are to be used.*

Proceedings Issues

For the individual papers to appear as are of a collection, such as a Journal or Proceedings volume, each paper must contain some things that can only be provided by the editor(s); for example, page numbers and running-heads or footers. For a collection of `.pdf` files, there also needs to be navigation back to a document which provides an overall Table-of-Contents, or other unifying material.

The driver and mini-driver setup makes it very easy to do this, with minimal editing within the individual manuscripts. Firstly the driver assigns a code-number to each job. This is done within `TUG99pdf.pre` by \TeX code that loops through all the values for `\jobname` until it finds a match with the current document, as follows:

```
\newcount\jobCode
\let\thisJobNum\relax
\edef\thisJobName{\jobname}
\edef\thisJobName{\meaning\thisJobName}
\loop\advance\jobCode by 1\relax
\getAuthorName{\the\jobCode}%
{\ifx\authorName\emptyJob
\gdef\thisJobNum{0}\fi
\edef\testjob{\authorName\revisionLevel}%
\edef\testjob{\meaning\testjob}%
{\ifx\thisJobName\testjob
\xdef\thisJobNum{\the\jobCode}%
\else \ifx ...
\else ...
\fi\fi\fi}}\relax
\ifnum\jobCode >50 \let\thisJobNum\emptyJob\fi
\ifx\thisJobNum\relax\repeat
```

where the `...` denotes extra code that copes with authors having written two or more papers. Notice the technical trick of using `\meaning`, to overcome differences in the category codes of letter-tokens in the expansions of `for` `\testjob` and `\jobname`. The macro-name `\thisJobNum` holds the required code-number after exiting from the loop, else is `\relax` if there has been some mistake (termination being guaranteed by the arbitrary maximum value of 50 for `\jobCode`).

The value for `\authorName` is supplied via:

```
\def\authorName{}
\def\getAuthorName#1{\edef\authorName{%
\ifcase #1\relax\or
fulling\or
ion\or
...
panelC\else\fi}}
```

in which the authors are listed within the `\ifcase` in the order that the talks were given, or will appear within the proceedings, or whatever other order is most convenient.

Now page-numbers or other things can be obtained from similar `\ifcase` listings; e.g.

```
\def\getTalkPage{\edef\authorPage{%
\ifcase\thisJobNum ???\or % something is wrong
  1001\or %fulling
  1006\or %ion
  1015\or %lovell
  ...
  1158\else
\fi}}
```

This is particularly convenient, as it is no longer necessary to set the page-number explicitly within each author's file, as was being done previously.

Similarly, the date and time scheduled for each talk were recorded in `TUG99pdf.pre`:

```
\def\getTalkDate{\edef\authorDate{%
\ifnum\thisJobNum=0 ??? % something is wrong
\else\ifnum\thisJobNum<10\relax Monday, 16%
\else\ifnum\thisJobNum<15\relax Tuesday, 17%
\else\ifnum\thisJobNum<25\relax Wednesday, 18%
\else\ifnum\thisJobNum<33\relax Thursday, 19%
\fi\fi\fi\fi\fi}}
```

```
\def\AM{\noexpand\,am}
\def\PM{\noexpand\,pm}
\def\getTalkTime{\edef\authorTime{%
\ifcase\thisJobNum ???\or % something is wrong
% Monday
  8.30\AM\or
  9.00\AM\or
  ...
  ...
  3.45\PM\else
\fi}}
```

This information was inserted automatically into the footer of each paper. Furthermore, the footer was made as an active hyperlink to the daily schedule, within `program.pdf`. Thus `program.pdf` serves as the wrapper, apparently combining all the papers into a single volume. A little bit of arithmetic was programmed to correlate the value in `\thisJobNum` with symbolic `\label` names used for anchors in `program.pdf`.

A significant advantage of using the driver file in this way is immediately apparent. Suppose the order of the papers is changed, a paper is withdrawn, or the page-lengths are modified. It is only necessary to make suitable adjustments within the driver file; the author's manuscripts need not be changed at all.

To \TeX or not to \TeX ⁴

Several papers for TUG'99 were submitted using \TeX , rather than \LaTeX . These were among the most troublesome to prepare for PDF. It is not difficult to adjust definitions of `\title` and `\author`

to cope with a different syntax. For example, a mini-driver `tug99tex.ltx` copes with the rudimentary book-keeping:

```
\input ../TUG99pdf.pre
\let\latextitle = \title
\let\latexauthor=\author
\let\latexaddress=\address
\let\latexnetaddress=\netaddress
\def\title *#1*{\latextitle{#1}}
\def\author *#1*{\latexauthor{#1}}
\def\address *#1*{\latexaddress{#1}}
\def\netaddress *#1*{\latexnetaddress{#1}}
\def\article{\begin{document}\maketitle}
\def\endarticle{\end{document}\endinput}
\def\head #1\endhead{\section{#1}}
\def\subhead #1\endhead{\subsection{#1}}
\def\subsubhead #1\endhead{%
  \noindent\textbf{#1}\ignorespaces}
\let\entry=\bibitem
```

```
\input{\jobname\revisionLevel.tex}
\end{document}
```

What is more difficult is to adapt or edit markup commands used within the body of the manuscript (in particular `\item` and `\itemitem`), or commands used for visual, rather than logical, effect.

Advice to Authors: *Please use \LaTeX ...*

It is not an issue of pride as to whether an author can typeset beautiful pages himself/herself, or that the default \LaTeX styles are ugly. Rather, it is imperative to recognise that the author is *not* in control of the ultimate page-layout and style in which his/her words will be typeset. \LaTeX 's main strength lies in the use of logical markup constructions within the body of the manuscript. This way the author's desires or intentions can be expressed, even when the implementation may be deficient or lacking altogether. Use XML, we can hear Sebastian saying.⁵

Advice to Authors: *...at least use \LaTeX -like markup syntax in the body of the document.*

The need for logical markup is even more imperative with the possibility of different types of output: author's manuscript on paper, printed preprint version, printed proceedings volume, electronic version in PDF and/or HTML. For example, the electronic interpretation of `\url` is very different, and much richer, than the interpretation for paper. Figures and tables should always be floated, no matter how much you detest using this for your own publications; layout is the editor's problem, not the author's.

⁵ He is not wrong; we just don't yet have enough robust tools or the experience with it to make this a convenient path to follow.

⁴ with apologies to Fred Bartlett [sic].

L^AT_EX, through its use of packages, already has well-defined markup conventions for just about everything that might appear in a manuscript. To not take advantage of this means that editors, in trying to give the richest possible interpretation for the particular medium, may not fully understand an author's intentions. This can result in outright errors, or delays in publication while an attempt is made to gain clarification. Instructions like “no macros” (which is clearly ludicrous for a journal about T_EX-related things) really mean “don't worry about the formatting, but logical markup is quite OK, provided we can change the definition to impose our own styles”. Since the latter is too hard to enunciate, and yet harder still to quantify, it usually comes out as “no macros” which is then largely ignored.

◇ Ross Moore
Macquarie University
NSW 2109, Australia
`ross@maths.mq.edu.au`