Production notes

Karl Berry

This seems an opportune place to say a few words about *TUGboat* production. In general, our process is nothing like as regularized as that described by Marcin.

One immediate difference is that *TUGboat*, by its nature, has to handle articles using any TEX engine. We use pdf(LA)TEX by default, which can handle the majority of articles, but it's typical and reasonable for an article about LuaTEX to require LuaTEX, etc.

So, we can't create an entire *TUGboat* issue in one run. Instead, each article is processed separately into its own PDF. We then concatenate the individual PDFs to make the full-issue PDF to be uploaded to our printer.

To do the concatenation, we've used a variety of tools, most commonly Ghostscript and `pdfjam` (`ctan.org/pkg/pdfjam`) of late. ConTEXt and `pdftk` have also been useful. Different tools are needed as years go by and software and systems change (for no convincing reason).

The same tools can select PDF pages when splicing two articles together, that is, when one article ends and another begins on the same page. We try to avoid this, partly because of the extra production trouble, but primarily because it is better for readers to find new articles starting on new pages. But content must dictate form, so we make it work out when it's needed. (Incidentally, another PDF check is for all fonts being embedded, using `pdffonts` from Xpdf, `foolabs.com/xpdf`.)

The trickiest part of producing the whole issue as a concatenation is the page numbering. We have a control file which lists all the articles in the order in which they will appear, as well as the beginning page number for the issue. Then each article writes its beginning and ending (`\AtEndDocument`) page numbers into external files, where the next article can read them. The two tables of contents use the same external files, so as to ensure consistency of the page numbers.

Unfortunately, nothing comparable keeps titles and authors consistent among the tables of contents and articles. Partly this is due to inertia, partly because it would be hard to implement in full generality, and partly because sometimes there are intentional differences among the three places — forced line breaks, abbreviations, etc.

Back to issue production: the compilation of each article, and the overall process, is done with GNU Make, via a single included Makefile fragment which defines nearly all needed actions. The per-article Makefiles merely give the name of the file, the engine to use (if not `pdflatex`), etc.; the goal being, naturally, to eliminate redundancy wherever possible.

We use GNU Aspell (`gnu.org/s/aspell`) with some `sed` preprocessing to do spell checking: `aspell list \ --mode=tex --add-extra-dicts=‘pwd‘/.dict.pws\ | sort -fu`. The idea being that a given article can have a `.dict.pws` file with the spelling exceptions needed that don't make sense to add to the global exception list (unusual proper names, one-off neologisms, etc.).

Besides spell checking, we've implemented several custom checks across an entire issue, again done in the central Makefile: doubled words (`math.utah.edu/~beebe/software/file-tools.html#dw`), lowercase letters inside `\acro`, tripled letters ("eee"), etc. More globally, we check that the tables of contents aren't missing an article processed in the central control file. Of course, besides the automated checks, humans review each and every word, line, and page that goes out.

Character encodings are an unending hassle. We receive many articles in UTF-8 these days, often with confusion or incorrect usage of accents, dashes, etc., or garbled in transmission. Other articles still use Latin-1 or similar. For articles which have only a few "special" characters, we strongly recommend taking advantage of TEX's inherent capability, and sticking to 7-bit ASCII.

One final point is that all production work is done on Unix (CentOS 7 these days), using TEX Live. Thanks to the well-known portability of TEX documents, there is rarely a problem with an author obtaining different results than the production run, with one glaring exception: when fonts are found by XETEX or LuaTEX via system lookup, instead of by filename. This makes the document immediately and completely unportable — so I implore everyone, please don't do this in *TUGboat* articles!