Abstracts — Bibliographies

Abusing TEX: `custom-bib` as an example

Patrick W. Daly
Max-Planck-Institut für Aeronomie
`daly@linmpi.mpg.de`

Although TEX is essentially a typesetting program, there are a number of “mis-uses” of it to accomplish what could be called off-topic programming.

The most complex example of this is no doubt the `fontinst` bundle, which creates the `.tfm` and `.vf` metric and virtual font files for PostScript fonts. Another service routine written in TEX with no `.dvi` output is `docstrip`, which is part of the kernel LATEX installation, and which is vital for that installation. Originally `docstrip` was intended as a utility to remove comments from installation source files, but it now contains an even more powerful feature: it can customize the output code according to preselected options, and it can combine code from several source files.

It was this property that I employed to simplify an old problem with BibTEX: that every publisher has its own list of arbitrary formatting rules, and it is not easy to write new `.bst` files to meet these demands. Thus I wrote a generalized *master bibliography style*, or `.mbs` file, originally providing some 50 options for alternative bibliography style points, to be converted to a `.bst` file with `docstrip`. Today, my `merlin.mbs` claims well over 100 options.

The more complicated part of the `custom-bib` bundle, however, is interfacing with the user to manage the myriad choices, and to generate a `docstrip` batch file to do the actual conversion. This required yet another pseudo-program in TEX language, `makebst`, which examines all the available options in the `.mbs` file, offers them to the user interactively, prepares the batch file, writes a protocol (for future changes of mind), and even runs the batch file. Without this, `merlin.mbs` would be totally unmanageable; it tames the wizard.

Such utilities written in the TEX language are guaranteed to run on all systems where TEX is installed. Any other programming language would involve problems of platform compatibilities and portability. This advantage outweighs the fact that as a programming language *per se*, TEX is a monster.

MlBibTEX version 1.3

Jean-Michel Hufflen
Univ. of Franche-Comté
France
`hufflen@lifc.univ-fcomte.fr`

In October 2000, we started a new implementation of BibTEX, the bibliography program associated with LATEX. This implementation is so-called MlBibTEX (for “Multilingual BibTEX”) because it includes multilingual features. Multilingual bibliographies can be organised with respect to two approaches:

reference-dependent approach each reference of a bibliography section of a document is expressed using the language of the entry: for example, the month name of a reference to a book written in English (resp. French, German, ...) is given in English (resp. French, German, ...);

document-dependent approach all the references of a bibliography section of a document are expressed using the document’s language, as far as possible.

After the first version (1.1), Version 1.2 provided more flexibility about the specification of names within the fields `AUTHOR` and `EDITOR`. Formatting such names in a bibliography section is easier, too. These two versions use the bibliography style language (`.bst`) of BibTEX and allowed us to define requirements for a new language for bibliography styles. The syntax of this new language is close to XSL-FO and this language will be used in Version 1.3. More precisely, the two languages will coexist in order to ease the transition between “old” styles and “new” ones. In the paper, we will:

- show how to design new styles with the “new” bibliography style language (it is completely described as an annex);
- explain how the coexistence between the two languages is organised.

(We expect to publish the full paper in the next regular issue of *TUGboat*. *Ed.*)