Abusing TEX: `custom-bib` as an example

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

Monday July 21, 2003

Abstract Although TEX is essentially a typesetting program, there are a number of "misuses" 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 uses his 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, containing 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 is the tamer of the wizard.

Such utilities written in the TEX language are guaranteed to run on all systems where TEX is installed. Any other program 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.