

A little UNIX *shell script* (the UNIX term for a command procedure) `palatex` might look as follows (the shell script has one argument `$1` which is the part of the document to be processed, without the file extension `tex`):

```
# UNIX palatex shell script
echo "\includeonly{$1}" > include.tmp
latex main
mv main.dvi $1.dvi
mv main.log $1.log
```

The above shell script would be, for instance, invoked as `palatex part1`. Two remarks with respect to this script at this point:

1. The name for the main source file in the shell script is fixed to `main.tex`. I normally store every document in a separate directory and I find it very convenient always to call the main source file by the same name. Naturally, the shell script could be modified to accommodate another parameter, which is the name of the main source file.
2. This idea can be applied even more conveniently in a `makefile`. Now all you have to type is `make`, and only those parts of the document which have to be processed will be processed. Here is such a `makefile`:

```
.SUFFIXES: .dvi .tex
FILES = part1.dvi part2.dvi\
        part3.dvi

all: $(FILES)

.tex.dvi:
    echo '\includeonly{$*}'>\
        include.tmp
    latex main
    mv main.dvi $*.dvi
    mv Main.log $*.log
```

Now let us write a batch file `palatex.bat` for MS/DOS along the same lines. In MS/DOS, `%1` stands for the first parameter in a batch procedure.

```
if exists %1.dvi del %1.dvi
if exists %1.log del %1.log
echo \includeonly{%1} > include.tmp
tex &lplain main
ren main.dvi %1.dvi
ren main.log %1.log
```

Observe that when using `palatex.bat` you type the name of the part *without* the file extension `tex`; so you type `palatex.bat part1` and **not** `palatex part1.tex`. If you type the file name with the file extension `tex` then you will have a catastrophe.

Assume you typed `palatex part1.tex`. Now the first line of the procedure expands to:

```
if exists part1.tex.dvi
    del part1.tex.dvi
```

(Observe the illegal file names formed this way.) MS/DOS will interpret this as:

```
if exists part1.tex
    del part1.tex
```

In other words, your source `part1.tex` is removed!

I am not an MS/DOS person but I encourage knowledgeable MS/DOS people to write a little C-program which aborts the command procedure if the name of a part was entered with a file extension. By the way, there is also a `make` program available for MS/DOS. Using that, you can automate the whole procedure under MS/DOS the same way I outlined it for UNIX.

Sue Volkmann from the *Woods Hole Oceanographic Institute* in Woods Hole, MA, helped me implement the ideas under MS/DOS.

## Page Layout in L<sup>A</sup>T<sub>E</sub>X: Erratum

Kent McPherson  
SLI Avionic Systems Corp.

In `LAYOUT.STY`, which was printed in Volume 9, No. 1, there were a couple of commented lines. Unfortunately, when the style file was printed in two-column mode, these commented lines were broken into two lines each, and the second line was then uncommented. This causes the macros to fail.

The lines in question appear in the middle of column 2, page 81. They look like:

```
%\omarginref=\omargin
    \advance\omarginref by \oneinch
%\advance\omarginref by \hofref
and
%\emarginref=\emargin
    \advance\emarginref by \oneinch
%\advance\emarginref by \hofset
```

These lines should be deleted from `LAYOUT.STY`.

Editor's note: We are sorry for any inconvenience that resulted from our omission of the comment character from the wrapped lines shown above, and wish to thank those of you who reported problems.