A brief guide to TEX assistants*

Sebastian Rahtz

ArchaeoInformatica
York, England
spqr@ftp.tex.ac.uk

This paper offers a 'back to basics' overview of the various types of software which are of assistance to the TEXnical writer on a personal computer (this includes Unix machines, but excludes, for instance, VMS or VM/CMS systems). It is based on a presentation to the October UK-TUG meeting, and I am grateful to the other speakers, and our chairman Allan Reese, for insights and news. I would also like to refer readers to the Dutch TEX Group's journal *MAPS*, whose issue 93.2 contains a variety of useful papers on TEX interfaces (two of which are reproduced in Baskerville Volume 3.2 (December 1993).

Every TEX user knows that the traditional command-line way of working (the 'edit ; compile; {preview, print}' cycle) is far from ideal, and many people want some help. We may define eight different ways in which the environment we interact with can be improved:

1. Integrated environments (e.g. TEXtures on Mac), where everything is tightly-coupled, and designed from the ground up to work together;
2. Integrating shells (e.g. TEXshell), where a set of different programs are controlled by a single interface, which does its best to protect the user from their idiosyncrasies and errors;
3. Loosely-coupled packages (e.g. Windows, Desqview setups), where a multi-tasking environment is used to combine together various programs to provide a useful working environment;
4. TEX-aware editors (e.g. Gnuemacs AUCTEX package), which simplify the input of complex markup, and check syntax;
5. TEX super-scripts, including Makefile generators (e.g. 'texit'); these take away the need to remember complicated program switches, and remembering which files are up to date, by understanding the rules about the different types of TEX-related files;
6. Super-previewers (e.g. xtex), which allow us to make greater use of the previewer to inspect and interact with our document.
7. Hypertext source convertors (e.g. latex2html), which make the tedium of writing generic markup with LATEX more worthwhile by giving it a life beyond the printed page;
8. Friendly TEX programming; when we *do* interact with TEX's enigmatic '*' prompt, we want macros to help us, not hinder us. This is discussed by Jonathan Fine in Baskerville Volume 3.2 (December 1993).

Since most computer users seem to belong in distinct 'camps', based on machines and operating systems, it seems appropriate to examine what is on offer in each area.

1 Windows, OS/2, and NT

For a fully-fledged Windows TEX, we have to look to the commercial companies: TurboTEX, from Kinch Computer Co, offers a full TEX, previewer, and drivers with a moderately-well integrated Windows interface, although (critically) it lacks a real integrated editor. The 'traditional' PC supplier of TEX, PCTEX, has also recently released a Windows setup, although it (like TurboTEX) fails to take advantage of builtin fonts properly. Y&Y sell a high-quality Windows previewer and PostScript driver, linked with a public domain TEX and commercial programmer's editor, which work together well in a loosely-coupled way, and offer genuine integration with the Windows font systems (although the previewer has trouble using PK fonts). *Scientific Word* takes a different direction by offering a WYSIWYG writers interface (particularly for maths) which saves files in LATEX format and does final formatting for printing with TurboTEX. None of these products, however, offer the same degree of integration and instant previewing as Textures on a Mac, and we still await the 'real' Windows TEX.

The more impoverished writer who wants to build her own Windows TEX setup has a not altogether easy time.[1] There is an excellent previewer, Hippocrates Sendoukas' *Dviwin* (now at version 2.81, but *still* lacking support for virtual fonts, and nowhere near using the Windows system fonts), and the excellent emTEX ports of the basic TEX programs run in DOS windows (with some shenanigans necessary to change extended memory managers), but integration is left to the user. The powerful editor Microemacs has an adequate Windows port, extensible to allow TEX jobs (compiling, previewing, printing — see Figure 1) and context-sensitive help to be selected from menus, but it does not provide much protection for the user from the essentially unsatisfactory nature of Windows DOS boxes and multi-

*Reprint from the Annals of the UK TEX Users' Group: **Baskerville**, Volume 3.2, December 1993.

[1] Wietse Dol discusses this more fully in Baskerville Volume 3.2 (December 1993).

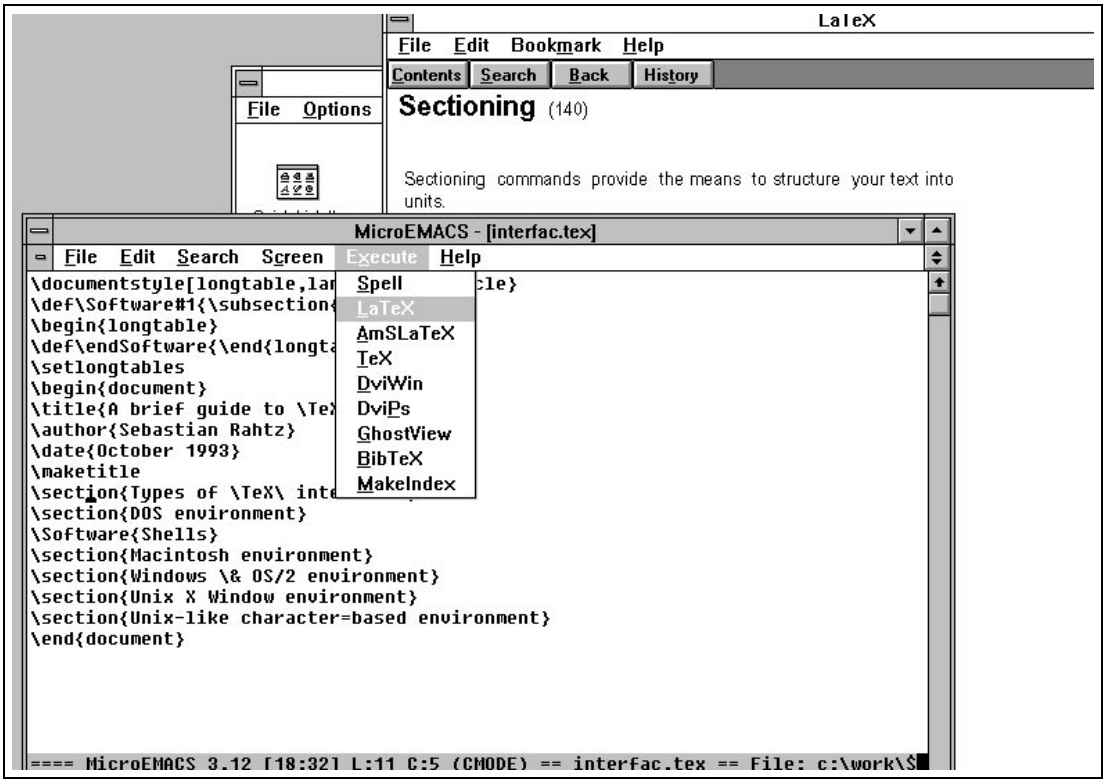

Figure 1: *Microemacs TEX menus*

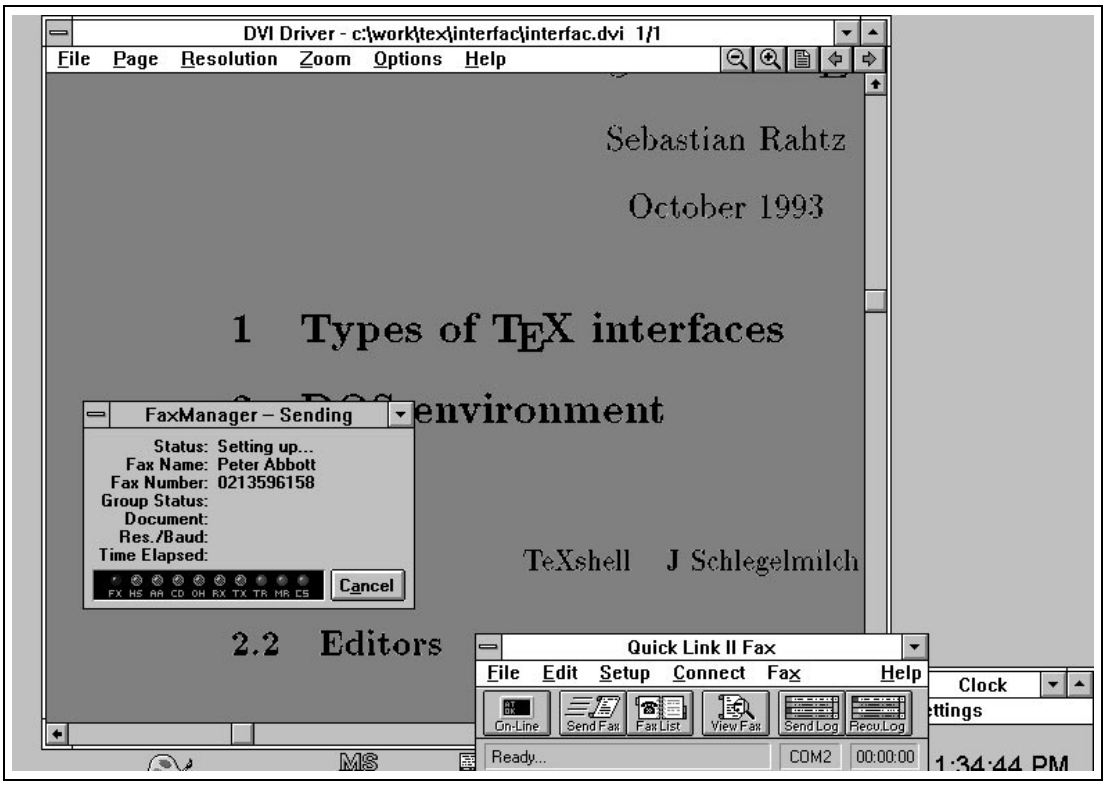

Figure 2: *Dviwin accessing Windows printer driver*

tasking. The alternative, running a DOS TEX shell in a DOS box under Windows, probably gives a better result (see Wietse Dol's article). But the convenience of access to Windows printer drivers (such as fax boards — see Figure 2) is a pointer to why this possibility should not be ignored.

Those converting to NT will, of course, have access to all the Windows tools, and the Unix 'web2c' TEX has been ported, but there do not yet appear to be any interfaces which make serious use of NT's abilities.

OS/2 users are well served by the basic emTEX kit, which has a proper Presentation Manager previewer, and by various shells (e.g. *PMTEX*) which make life easy. Given the good multi-tasking, Windows compatibility, and the availability of compilers to port most Unix tools (such as *Gnuemacs*, *make* etc), the OS/2 TEXie has the best of almost all worlds.

2 DOS

The hard-pressed DOS user (or lover — there are a few) has a plethora of tools to make up for the inadequacies of the command line. The most mature of the 'shell' programs, which run TEX, previewer, printer, editor etc for you sequentially when you select menu items from a character-based screen, is probably Jürgen Schlegelmilch's TEXshell (see Phons Bloemen's article below). A typical screen is shown in Figure 3. Other offerings include Thomas Esken's TEXsurface, Johannes Martin's TEXpert, Ulrich Jahnz's Eddi4TEX, the editors Redit and ET (see Reese's article above) and Petr Olsak's menu builder 'mnu'. All of these have some or all of the following characteristics:

- a model of the original Turbo Pascal — character screens with pull-down menu bars;
- protection from the horrors of DOS environment variables controlling programs from four different authors (*after* you have got the shell properly configured);
- TEX and other programs activated by 'shelling out' to DOS, (an intrinsically 'uncool' procedure), with parsing of error messages;
- heavy use of customizable environments (see Figure 4), which can confuse the novice to screaming point;
- limited editors which can be difficult to use with 'industrial strength' documents;
- access to online TEX context-sensitive help.

This is not to deny the great usefulness of parts of all of the DOS TEX shells, but their parts are greater than their sum. The dedicated DOS TEX user will find them extremely useful, but perhaps they can be summed up by saying that they will not convert many Microsoft Word users.

A recent initiative by Dutch TEX users has resulted in 4TEX (see the article by Wietse Dol), a TEX shell accompanied by a packaging of almost all the good public domain TEX goodies. This is undoubtedly the system which is getting most support and development. The even more powerful AsTEX setup of Michel Lavaud (described in *TUGboat* 14.3, 238–44) unfortunately depends on commercial software for its implementation, but those interested in a wider perspective on *managing* their TEX document library should examine this closely.

Those who want the approach of Scientific Word (WYSIWYG maths composition) in plain DOS can try the public domain ET by John Collins, or the commercial *Leo* from ABK Software.

3 Macintosh systems

The Macintosh user does not need convincing about GUIs or integration; for hard cash, she can buy what is probably the fastest (given the processor), slickest TEX around, Textures. A port of the Unix TEX is available (CMacTEX), and a half-way house, the shareware OzTEX, which offers integration between a good compiler, previewer and printer driver, but not quite the smooth transition of Textures, or the use of system fonts. What is lacking on a Mac is a WYSIWYG interface to mathematical writing such as that offered by Scientific Word.

4 Unix, etc

The Unix TEX user has a rich variety of tools to play with; like everything else in Unix, they are dangerously powerful, and require dedication to obtain true mastery. They start with simple things like the *csh* or *bash* history mechanisms, which let you recall previous commands by name:

```
$ history
  171  lutex fine2
  172  dvips fine2
  175  latex interfac
  176  lpq
  180  dvips fine2 -o fine2.ps
  182  cp fine2.ps /user
$ !lat
$ !dv
```

Another universal Unix tool is 'make' which manages the relationship between different stages of a document's 'compilation' from various sources, and keeps things up to date. A large book, with dozens of input files, hundreds of graphics inclusions from different programs, a bibliography managed with BibTEX, and an index, is likely to need such a tool to keep the author sane. There are various tools to help you write 'make' files, and to do similar jobs to 'make':

imaketex Just write an *imake* file and you just type 'make'; as in 'just write'...

texit A Perl script which makes educated guesses about the state of your TEX file and helps you decide what to do next;

latexmk Given a LATEX file this tries to work out what to do, like a 'makefile'; but does not understand multiple-level dependencies;

latex_make A set of rules for real 'make', which understand a wide range of common TEX jobs; but tailoring is not for the beginner. Sample 'make' rules are shown in Figure 5.

If you are already a Unix programmer, these are well worth investigating.

There do not appear to be any obvious public domain TEX shells for the X Window system; the 'doc' editor does provide help in writing LATEX, with limited immediate formatting, but it still leaves you to run TEX. The Berkeley VorTEX program to produce an interactive TEX ended some years ago without a real success (although various results

Figure 3: *TEXshell, after running TEX and finding an error*

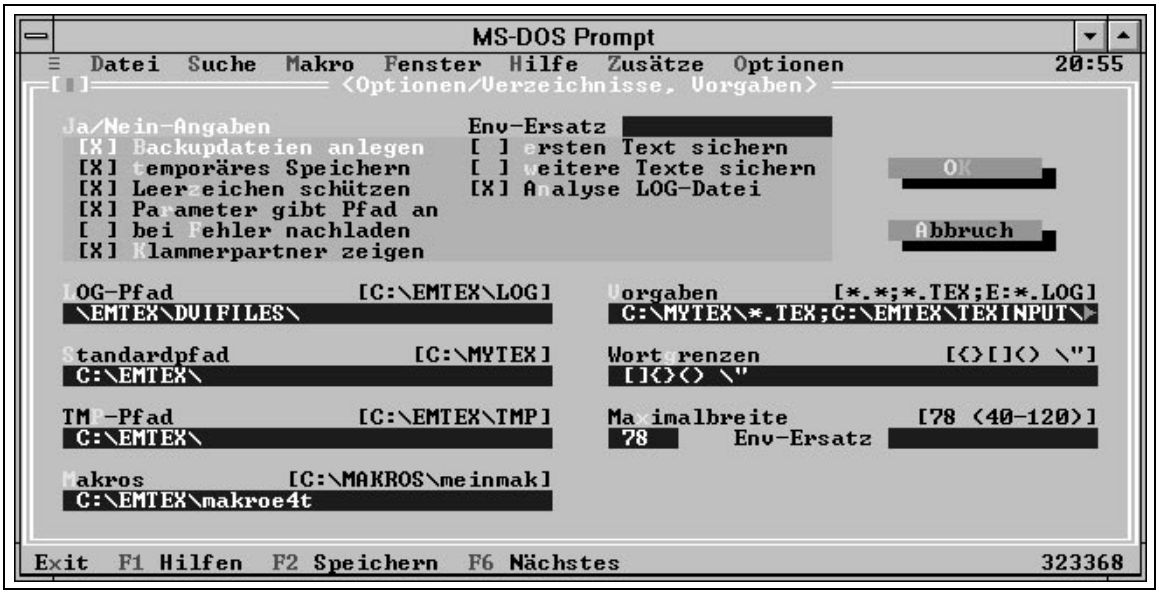

Figure 4: *Eddi4TEX customization*

have now been released for general use; they are available in CTAN).

The queen of Unix software is Gnuemacs; as well as being an *extremely* sophisticated editor, it provides hooks to run jobs synchronously or asynchronously from within the editor, and parse the output. The latest X Window version allows for user-defined menu bars, and multiple windows, which allows the Unix user access to as sophisticated a setup as almost everything except Textures. The actual editing can be enhanced with powerful Lisp routines to check syntax provide structure skeletons, edit in outline mode etc; the best-supported and enhanced package is Kresten Krab Thorup and Per Abrahamsen's AUCTEX. If you can learn to be happy with Gnuemacs, this has almost limitless possibilities. An example screen is given in Figure 6.

5 Conclusions

If you have to choose a TEX environment, how do you start evaluating the offerings? I would suggest that base your choice on the following criteria:

- The editor; this is where you spend most of the time;
- On-line help; LATEX is unmemorable;
- Integration with windowing systems; doing just TEX is dull;
- *Lack* of special features; compatibility with other TEX setups is important;
- Integration with the 'style' of the environment (i.e. use of Adobe Type Manager).

```
# ensure that the bbl file gets regenerated if the bib file is changed
%.bbl : $(BIBFILES)
        @# if there is no aux file, skip this, it will get done later
        -@if [ -r $*.aux ] ;\
        then $(BIBTEX) $* ; \
             fi
# create a dvi file from a tex file
% %.dvi: %.tex
        $(LATEX) $*
        -@egrep -c 'Citation .* undefined.' $*.log && ($(BIBTEX) $*;$(LATEX) $*)
        -@grep 'Rerun to get cross-references right' $*.log && $(LATEX) $*
# create postscript file of a dvi file
%.ps: %.dvi
        echo dvips -D $(PRINT_RES) -p $(PAGE_F) -l $(PAGE_T) -o $@ $*
        dvips -D $(PRINT_RES) -p $(PAGE_F) -l $(PAGE_T) -o $@ $*
```

Figure 5: *An example of 'make' rules for a TEX file*

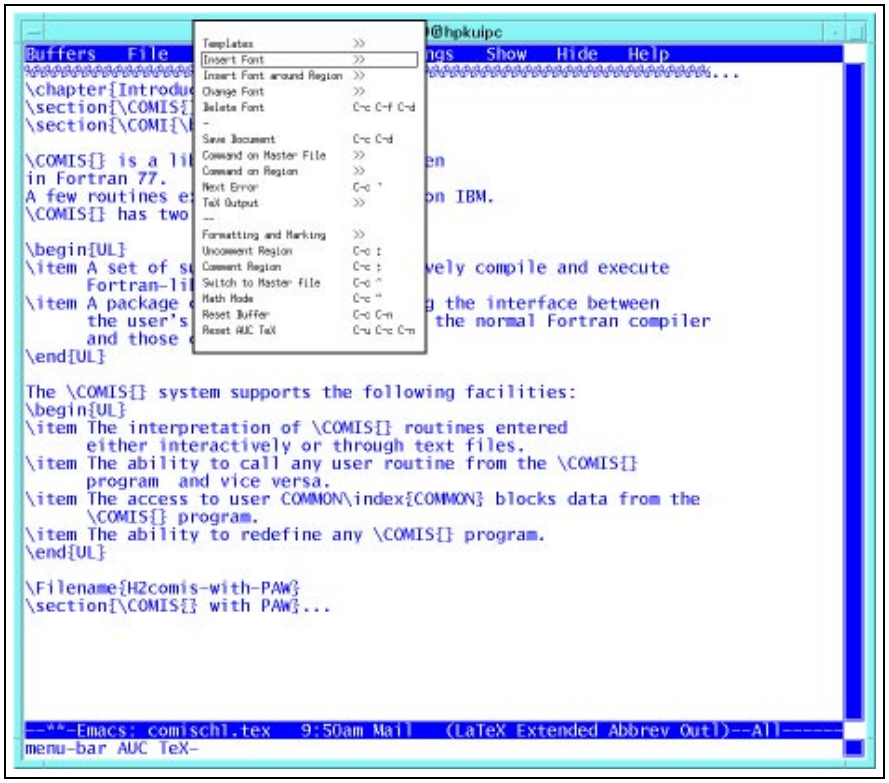

Figure 6: *Gnuemacs and AUCTEX*

For the five possible environments you might consider choosing:

- AUCTEX: if you use the X Window system and *gnuemacs*, there is no other way to live;
- TEXshell: if you like plain DOS, this is well-designed and easily customized;
- Dviwindo: it really is a Windows application, and understands PostScript fonts;
- TEXtures: it works; it has semi-instant previewing; it integrates its fonts with the rest of the Mac;
- OS/2 and emTeX: a very good command-line TEX, a Presentation Manager previewer, and integration with a reliable operating system.

[Note: all the public domain or shareware software referred to in this article can be found in the CTAN archives. The commercial vendors all advertise in TUGboat.]