

## Site Reports

### TEX-L Access for Bitnet Users

Glenn Vanderburg  
Texas A & M University

Texas A & M operates a TEX file- and list-server for those with Bitnet access. To use the server, send commands to LISTSERV@TAMVM1. Commands can be sent either interactively or in mail files. To get a list of the available TEX-related files, send the command "get tex filelist". You can then get any of the files listed with "get filename filetype". The server accepts many other commands; issue an "info" command to find out more.

The list of available files is currently small; TEXhax archives and files, WEB sources, and the LATEX style repository. We hope eventually to offer most of the files that those with ARPAnet access can get from Score (sources, change files for various systems, macro packages and documentation, etc.). Suggestions are welcome, and should be sent to X230GV@TAMVM1.BITNET (along with complaints and problems). And if anyone would like to donate a disk pack or two, that would help out a lot!

Finally, TEX-L is a TEX discussion list and TEXhax redistribution. To subscribe, send this command to LISTSERV@TAMVM1 with your name: "subscribe tex-l firstname lastname" and you will begin receiving TEX-L and TEXhax from a nearby Bitnet list server.

### TEX and Training

#### A Case Study

Laurie Mann  
Stratus Computer  
Marlboro, MA

*Over the last seven years, many "Site Reports" have appeared in these pages. This is a site report with a difference — I'm not a programmer or a system administrator. This is a look at the issue of TEX training at Stratus, and how we hope to improve training for new TEX users.*

For the last 3 $\frac{1}{2}$  years, Stratus Computer has been using TEX to produce software manuals and

other publications. Between 1981 and 1983, Stratus had published about twenty manuals, with all the production work being sub-contracted out. But Ron Webber, the Publications manager, wanted to bring production in-house. He looked into several production systems, and decided to port TEX to the Stratus and use the Compugraphic 8400 as the output device. With some help from Eric Janszen, he ported TEX, wrote a driver, wrote a format file and taught four technical writers the basics about formatting manuals with TEX.

I was hired back in November, 1983 as the first Publications Specialist. The first few weeks were very interesting and very frustrating. My previous computer experience was pretty limited. I knew how to log in. I knew how to use Emacs. Ron spent some time the first day getting me acquainted with the Stratus equipment, and giving me some introduction to TEX. But being fairly non-technical, I found everything very confusing. It took me a while to remember that a file was not permanently changed **until** I'd written it out. I had difficulty comprehending the differences between executing a command macro and working in an Emacs buffer (something I hate to admit today).

Some of TEX looked easy. Before Ron hired me, he showed me the format file he'd written, and some of that even read like English. Besides, at first I was only supposed to be debugging TeX files that the writer had formatted. Compiling and binding programs is easy, right? But debugging TEX files is a not a trivial activity. I remember the first time I saw the error,

```
TeX capacity exceeded, sorry
If you really absolutely need more capacity,
you can ask a wizard to enlarge me.
```

I was absolutely lost. Ron explained what most of the error messages meant, including some I'd seen that he'd never seen when he was experimenting with TEX. But, over the winter, the error messages became less frequent. And I generally ceased to panic when I did see an unfamiliar error. I started to learn how to format manuals, so the writers could concentrate on writing rather than formatting. Formatting, particularly the challenge of producing nice tables and usable indexes, really appealed to me.

I read everything I could about TEX, but there really wasn't very much material available. I had a line-printer copy of *The TEXbook*. While the first few chapters were absolutely invaluable, most of it seemed to be aimed at programmers. Ron's written introductory material on TEX read the same way.

Though I struggled with  $\text{\TeX}$  a lot as I was first learning it, within a few months I was  $\text{\TeX}$ nician enough to format manuals and send off completed manuals to our printer every month.

The fledgling production group added a second book producer in April, 1984. Carol Klos also had little computer experience before coming to Stratus, but was interested in learning about manual production. She also found learning  $\text{\TeX}$  difficult, but was soon producing high-quality manuals. That fall, I became the Production group's manager. By mid-85 there were five Publications Specialists and one full-time Typesetter in the Production group. We also added a Compugraphic 8600 typesetter and an MCS High Speed Processor. The department produced something over sixty manuals, some newsletters, and some long Marketing projects in 1984 and 1985.

One of first things I was supposed to do upon becoming the group's manager was to assemble an in-house production manual. Well, one thing led to another, and the manual sits in assorted pieces in my directories. Over the last few years, I restructured Ron's basic  $\text{\TeX}$  information and included lots of examples. I worked with Mary Fusoni, the second technical writer at Stratus, to develop formatting and indexing guidelines for writers. I wrote up many brief descriptions of how to use tools and equipment, but this information is disorganized and is sometimes later found to be wrong (I've since strongly encouraged people to edit what I write). While I'm a  $\text{\TeX}$  resource, I'm not a very good teacher, and the training materials we have are somewhat lacking.

Back in 1985, when three new people joined the group in a six week period, I worked with each person individually for a time but pretty much encouraged her to learn things on her own. The worst mistake here was giving new hires too much, too fast. What I should have done instead was to introduce new employees to the computer system, and have them spend a few days learning Emacs and basic operating system (VOS) commands. I should have waited before introducing  $\text{\TeX}$ , working with the batch queue, debugging files, using the phototypesetter, etc. But despite the fact that everyone had early difficulties, the entire group has now been together for over two years. For the last two years, manual production has been so efficient that manuals were completed and printed before the software release tapes were cut.

In-house manual production saves time and money. Not long after Ron brought Production in-house, he created positions for Editors and Project Leaders. He also had us strengthen department

procedures, and develop a team approach to manual publication. Our department now has five project leaders, each one responsible for an area of Stratus manuals (Communications, End-User, Languages, etc). Each team consists of a Project Leader, Writer, Editor and Publication Specialist. The Publication Specialist is responsible for most of the formatting, illustrating, and production of the manuals. The Writer writes the manual, but does minimal  $\text{\TeX}$  formatting. The Editors and Projects Leaders usually don't touch the source files at all, and work only on typeset galleys. We all believe this team approach has helped us improve the quality of our manuals. Our customers have been giving us good marks over the last two years. Three manuals were published in 1985 won local Society for Technical Communication (STC) awards, and one of those took an International STC award.

Ron was a member of TUG in the early '80s, and I joined in 1986, after spending a year in the STC. But our only contact with TUG was through TUGBOAT. When I went to the  $\text{\TeX}$  conference at Tufts last summer, I wasn't sure what to expect. Would I be the only non-programmer there? Was  $\text{\TeX}$  training an issue for other sites? It turns out that **all kinds of people use  $\text{\TeX}$** , from secretaries to technical writers, from graphic designers, to, yes, computer programmers. A number of us got together to discuss how to improve  $\text{\TeX}$  training, and we all felt it was something we should pursue in the future.

Over the winter, I came to the conclusion that as much as I liked working with  $\text{\TeX}$ , I really didn't much like the managerial side of the job, so I switched jobs with Carol. This has given me more time to study  $\text{\TeX}$ , as I've been implementing a new format file. Additionally, I have started to organize material for that production manual. This is important now because Carol will be hiring more Production Specialists over the next few months. Carol and I want to take a more structured approach to the training of new people. Additionally, we are going to see if we can hire people with prior publishing and/or computer experience (send your resumes to Carol if you are interested—we may also be hiring a tools programmer later this year).

While hiring non-technical people to work with  $\text{\TeX}$  succeeds in the long run, it can cause a lot of job frustration, particularly in the first months. There are several reasons for this:

1. Insufficient instruction on how to use VOS.
2. Insufficient instruction on how to use Emacs.
3. Insufficient material on basic  $\text{\TeX}$  information for non-programmers.

4. Insufficient organized material on formatting standards and production tools.

Stratus now has good introductory manuals about VOS and about Emacs. We will need to help new people get adjusted to using computer equipment, and have taken this into consideration as an important part of new-hire training. Until new hires are comfortable using VOS and Emacs, it's counter-productive to present much about T<sub>E</sub>X.

We want to take a more structured approach to teaching production skills in general and T<sub>E</sub>X in particular. Initially, only the basics about T<sub>E</sub>X will be presented. It's been my personal experience that overloading T<sub>E</sub>Xnical detail early on is very confusing to most non-programmers. (It's very interesting once you've adapted to using T<sub>E</sub>X, but not immediately.) Once the new hires master simple T<sub>E</sub>X concepts (like font designations, spacing, running T<sub>E</sub>X) they can go on to learn about how T<sub>E</sub>X is used to create "beautiful books."

Since last summer, with the help of others in the Publications department, I've been assembling an outline for the production manual. Actually, the planned manual is now two—a tutorial and a reference manual. Here's the proposed outline for the tutorial:

#### Introduction

#### Section 1: VOS Tutorial

- A. VOS and Stratus Computers
- B. Using VOS
  1. Logging-In and Logging-Out
- C. Using Emacs
  1. Moving the Cursor
  2. Moving Text
  3. Saving Text
- D. File Management
- E. Basic VOS Commands

#### Section 2: Introductory T<sub>E</sub>X Reference Material

- A. Basic Printing Concepts
  1. Fonts
    - a. Characters
  2. Spacing
  3. Page Make-up
- B. Typesetting Languages and T<sub>E</sub>X
- C. Stratus Implementation of T<sub>E</sub>X
  1. Type Fonts and Type Faces
  2. Point Sizes
  3. Spacing
    - a. Examples of Vertical Spacing
    - b. Examples of Horizontal Spacing
    - c. Special Characters
  4. Running T<sub>E</sub>X
    - a. Debugging Log Files
    - b. Preparing Files for the Typesetter

#### 5. Introduction to Machines

#### Section 3: T<sub>E</sub>X Tutorial

- A. Exercise 1: A Letter
- B. Exercise 2: A Sign
- C. Exercise 3: A Phonelist
- D. Exercise 4: A Flyer
- E. Exercise 5: A Form
- F. Exercise 6: A Table
- G. Exercise 8: A Report

*Note on presentation of samples: Each exercise will begin with the specifications for a job and an unformatted file (the manual will also direct the reader to an on-line, unformatted file). In the back of the book, the final, formatted file will appear on a left-hand page and the typeset output will appear on a right-hand page.*

Appendix A: Formatted file and typeset output for each exercise

When people are brought in, they will have five experienced Publications Specialists to help train them. We also hope to have this tutorial in place in the next few weeks, with the reference manual to follow later in the summer. I'll let you know how things go.

#### Data General Site Report

Bart Childs

We have updated to T<sub>E</sub>X 2.1 and now support DG's 4558 laser printer, which is a pretty plain Canon engine. I mentioned some of its limitations in the last issue. We seem to have little or no problem with it except that the size of the font downloading files is inordinately large. It can still operate near rated capacity with parallel or 19.2k serial interfaces.

We also have graphics inclusion. The source of the driver is the same as for our QMS driver; thus we are making it by use of a change file and have a large number of "device dependencies" as well as the "system dependencies". We reserve the latter for the operating system and compilers.

Graphics inclusion in the QMS uses the QUIC language and Canon-compatible instructions on the 4558. We use QIC and LBP for the file extensions. I have written a WEB code for the translation of

QIC files to LBP files. It is limited at this time to translation of bitmap patterns.

We require that the  $\TeX$  sources allocate the space for the graphics to be included. The graphic named `drawing` is included by use of the  $\TeX$  command:

```
\special{copy{drawing}}
```

This implies that `drawing.qic` is read by the QMS driver and that `drawing.lbp` is read by the 4558 driver. This is a step toward the goals mentioned by Leslie Lamport in the last issue without committing to PostScript only.

I suggest that we introduce a companion file with the extension `.siz` that would be created by any graphics-producing program or translator. This would give  $\TeX$  the information necessary to allocate the space for graphics.

## Enhancements to $\TeX$ on the VAX

Adrian F. Clark  
British Aerospace

This note describes two enhancements to  $\TeX$  under VAX/VMS. These are believed to be equally applicable to the two VMS implementations, by David Fuchs (the Stanford distribution) and by Kellerman and Smith.

The first enhancement allows  $\TeX$  to determine automatically whether or not it was invoked by an interactive user. If it was,  $\TeX$  runs in `error_stop_mode`, where it will prompt the user for instructions if an error is detected. When used non-interactively,  $\TeX$  runs in `batch_mode`; errors are recorded in the transcript file but execution is not affected. This facility was absent from the last Stanford distribution received by the author ( $\TeX$  1.3) while Kellerman and Smith use a command line qualifier to set batch mode.

The interactive or batch nature of a job is determined by testing whether the process-permanent logical name `SYS$INPUT` translates to a terminal device. (`SYS$INPUT` is used because of the prompting mechanism of `LIB$GET_FOREIGN`.) This means that, when executed from a command procedure by an interactive user,  $\TeX$  runs in `batch_mode`; this is similar to other VMS utilities. The technique works for "virtual" and networked terminals as well as

directly-connected devices. It has been tested over both DECnet and VAX/PSI (X-25) links.

The second enhancement is to interface  $\TeX$  to an editor, a feature which is missing from both VMS implementations. When  $\TeX$  spots an error in an input file and prompts the (interactive) user with "?," a reply of "E" invokes the editor and positions the cursor at the place in the input file where  $\TeX$  spotted the error. After correcting the text, exiting the editor causes the file to be written to disc; the context of the error is again displayed and the "?" prompt re-issued.

The editor which has been interfaced to  $\TeX$  is callable TPU, since it allows the cursor to be positioned at the appropriate part of the file with ease. However, the user is not restricted to the default EVE editing interface. Customised editors, such as the EDT emulator, can be selected in the usual way, simply by assigning the appropriate section file to the logical name `TPUSECINI`.

To position the cursor at the correct point in the text, a file is created in `SYS$SCRATCH` containing the required TPU commands. This is executed as the editor starts up and deleted as it exits. It is possible to add commands to this initialisation file; this could be used for, say, loading  $\TeX$ -related keypad definitions. To do this, the TPU commands should be stored in a file and its name assigned to `TeX$TPU_INI`. The contents of the file are then written into the TPU initialisation file before the cursor-positioning commands.

As an example, the author uses the `EVEplus` editor (available through DECUS) with an extension to provide an EDT-like keypad layout. The commands to set this up are (on the author's VAX):

```
$ DEFINE TPUSECINI  BAE$SYSTEM:EVEplus
$ DEFINE TeX$TPU_INI SYS$LOGIN:EVEplus.INI
```

where `EVEplus.INI` contains the line:

```
set_edt_keypad;
```

The enhancements are largely made via  $\TeX$ 's VMS change file, although a little of the code is contained in a separately-compiled file written in Fortran. The change file is based on David Fuchs'  $\TeX$  1.3 implementation, modified by the author for  $\TeX$  2.0. The interaction-testing code works on any VMS 4 version; the editor interface was written under VMS 4.5 but should work on VMS 4.2 and later systems. The files which implement these enhancements are available to the  $\TeX$  community; they can be obtained, for those with access to JANET, by MAILing a request to `UK.AC.KCL.PH.IPG.:ALIEN`.

## IBM VM/CMS Site Report

Dean Guenther  
Washington State University

Back in January, a new IBM VM/CMS distribution tape was made available through Maria Code. This tape corrected one problem in the CMS version of  $\TeX$ . The problem showed itself in documents which used `\write` statements. Specifically, a `\write` with an argument of more than 256 characters was being truncated. The max length has now been increased to 2048.

On this tape were extensive corrections to the drivers for the IBM 3812, 3820, 3800-3, 4250, and APA6670 (Sherpa). Thanks for these are due to Bob Creasy. These drivers now can imbed inline GDDM or SAS/Graph graphic page segment. For example,

```
\special{IPS filename}
```

where *filename* has the filetype of PSEG3820. Refer to the following article by Gil Pierson on creating SAS files for imbedding into  $\TeX$ . Thanks to Gil for getting this going.

Another contribution from Bob was a utility, PFS, which creates the Computer Modern font set for the 3812, 3820, 4250, APA6670 or 3800-3. It uses standard METAFONT and sets the blackness and other parameters depending on which printer it is creating fonts for. It does take quite a bit of CPU time, so I would recommend that you use VM/Batch for this job.

Agnes Hsu added the missing filemode specification in the CMS changes file for BIBTEX.

The CMS version of METAFONT now accepts information from the command line. Corrections were made to these MF files: CMBASE, CMTT12, CMTT10, CMR5, GREEKU, CMSS9, and CMSSI9. This brings these 7 files up to date. A new MF file, CMHINCH (half inch) has been included for those printers that cannot handle CMINCH.

Other changes:

- A utility to convert old 2K TFMs to the new 1K format has been added;
- The changes file for INIT $\TeX$  is now included;
- Faster versions of the CMS  $\Rightarrow$  MVS transfer programs (APART/TOGETHER);
- Finally, GFtoDVI now works properly on CMS.

The same week this tape was sent to Maria, the message "It happened..." came across the network. (See TUGboat, Vol. 8, No. 1, pg 6.) I didn't have time to install the fixes to make  $\TeX$  2.1 until April. Along with  $\TeX$  2.1, this tape contains a CMS changes file for GFtoPK. The first tests of

GFtoPK seem to work fine. I also modified the PSIZZL macros to use CM instead of AM fonts. Finally, the file for testing IBM fonts (IBMFONT.TEX) was fixed. As of May, Maria has the VM/CMS  $\TeX$  2.1 distribution.

As for IBM font tapes, by the time this is published, Maria will have MVS font tapes for the IBM 3800-3, 3820, and 3812. Maria also has CMS dump format tapes for the 3800-3, 3820, 3812 and 4250. No font tape is available for the APA6670.

Janene Winter has put together a collection of *mode\_def* settings that she has found to be optimal for the IBM printers cited above. These settings appear along with comparable settings for other printers in the overview beginning on page 132.

## SAS Merged with $\TeX$

Gil Pierson  
Washington State University

If you want to merge SAS/Graph output into  $\TeX$  output on the IBM printers, you can do so fairly easily with the VM/CMS distribution driver programs. First you build a SAS program with the GDDMNICKNAME of IBM3820. Specify the GSFNAME to be the filename of the input file. The GDDMTOKEN must reflect the resolution of the printer, in this case IMG240X. Consider this example which uses the filename of DRUM3:

```
OPTIONS NOTEXT82;
GOPTIONS DEVICE=GDDMFAM4 GSFNAME=DRUM3
GSFLEN=264 GSFMODE=REPLACE
GDDMNICKNAME=IBM3820 GDDMTOKEN=IMG240X
NODISPLAY;
DATA A;
  M=2; N=5; PI=3.1416;
  DO Y=0 TO 1 BY .02;
    DO X=0 TO 1 BY .02;
      Z=2*(SIN(M*PI*X)*SIN(N*PI*Y)+
          SIN(N*PI*X)*SIN(M*PI*Y));
      OUTPUT;
    END;
  END;
FORMAT Z 4.1; RUN;
PROC G3D;
PLOT Y*X=Z / CTOP=GREEN CBOTTOM=RED;
TITLE C=WHITE F=SIMPLEX 'M=2 N=5';
```

Next, you must create a PROFILE ADMDEFS file which must have the filename included, in this case DRUM3.

```
NICKNAME NAME=IBM3820,
  TONAME=(DRUM3,PSEG3820),
  PROCOPT=((CDPFTYPE,SEC),
           (HRIFORMT,CDPF),
           (HRISPILL,NO),(HRISWATH,1),
           (HRIPSIZE,30,30,TENTHS),
           (PRINTCTL,6,0,1,66,0,0,0))
```

Then call SAS to create the file:

```
GLOBAL TXTLIB ADMGLIB ADMPLIB
SAS DRUM3
```

The file created is DRUM3 PSEG3820, which can be included with a `\special`. It has the form,

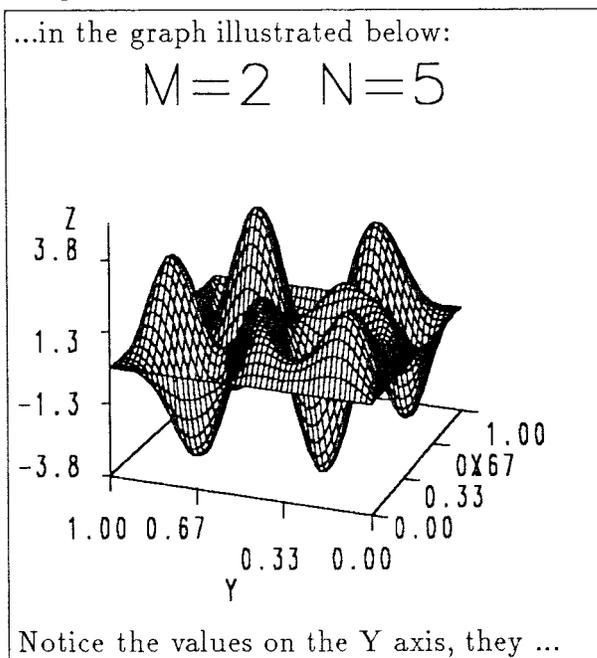
```
\special{IPS filename}
```

where *filename* is DRUM3 in this case. For example,

```
\noindent
...in the graph illustrated below:
...in the graph illustrated below:
```

```
\special{IPS drum3}
\vskip 3in
\noindent
```

Notice the values on the Y axis, they ... will print as:



The above sample was printed on a 240dpi IBM 3820.

## Unix $\TeX$

Pierre MacKay  
University of Washington

The most significant news for this Unix  $\TeX$  site report is that a System V port for the AT&T 3B2 has been completed by Lou Salkind of NYU, and is now part of the regular Unix distribution. This port depends on the System V version of the BSD4.x pc compiler, and is still limited by the fundamentalist rejection of the default, or "others" case in Pascal case statements. It is therefore necessary to find a source for ppx (the BSD4.x Pascal source prettyprinter) and to make use of the still undocumented `-O -f` flags which wrap up every case statement in a set-inclusion test. We can now hope that other machine-specific System V ports will follow. It is not clear yet whether the version of Berkeley pc will migrate to non-AT&T systems or whether we should look for new ports perhaps developed with commercially available Pascal compilers. In either case, the System V barrier is at last down, and the other half of the Unix world can now look forward to the availability of  $\TeX$ .

A second approach to System V  $\TeX$  is Pat Monardo's Common  $\TeX$  in C, being developed at Berkeley. Working versions of this are already known at a number of sites, and it is currently being upgraded to correspond with  $\TeX$  Version 2.1. Common  $\TeX$  is coded in traditional C fashion, without the rich documentation of a WEB source, and without the convention of a pool file, but on many architectures it is likely to produce somewhat tighter and faster executable code. It is not yet a part of the Unix  $\TeX$  distribution.

The move from BSD4.2 to BSD4.3 on VAXen has caused very few serious difficulties. Recompile is necessary owing to the change in the format of a.out files, which also makes it impossible to use the old version of undump.

On Suns, the changes are more radical. There is a large gulf between Sun operating system 3.x and its predecessors. This is further complicated by the difference between Sun-2 and Sun-3 versions of the new O.S. Sun-3 users (68020 CPU) have no serious trouble with any version of O.S. 3.x, but Sun-2 users must upgrade to O.S. 3.2 before trying to compile  $\TeX$ . It is a measure of the breadth of the Unix  $\TeX$  community that I first learned from Peter Iliev in Scotland of some possible improvements in the Sun compilation. The pc that comes with the new Sun O.S. has adopted the "otherwise" heresy, and provides some other nice features not formerly

available. Moreover, the "optimization" loop in the assembler has been cleaned up, and compilation no longer takes a week of elapsed time. It is now possible to compile the output from tangle directly, without the use of either ppx or the old sed script for breaking up the source file.

We are still working to track down ports for Celerity, Sequent, Intelligent Systems, Gould and others.

Output drivers have remained stable. (Too stable in the case of the dvi-to-PostScript driver.) The best news is that Matt Thomas has rewritten the LN03 driver to support all the varieties of METAFONT output. His driver is now the standard offering on the distribution tape. A good deal of work has recently been done on dvi-to-PostScript drivers. Richard Furuta is presently looking at one possible upgrade developed in Canada, and I am looking at the Unix relevant parts of Nelson Beebe's collection of drivers available from Utah. (The Utah drivers are accompanied by a thorough manual which can be read with profit by anyone concerned with dvi output.) We hope to improve the T<sub>E</sub>X devices directory in the summer by incorporating a good part of the work mentioned here. We hope also to offer the first genuinely open access to a true high-resolution phototypesetter before the year is out.

### Contents of the distribution

Current versions of standard programs in the distribution:

(in the `./tex82` path)

- T<sub>E</sub>X 2.1 (implying use of the new cm fonts) (plain.tex version 2.3)
- L<sup>A</sup>T<sub>E</sub>X 2.09 (release of 19 April, 1986, with corrections to 22 February, 1987) (also SliT<sub>E</sub>X 2.09)
- tangle 2.8
- weave 2.9
- dvitype 2.8
- pltotf 2.3
- tftopl 2.5

(in the `./mf84` path)

- METAFONT 1.3
- chtopx 1
- gftodvi 1.7
- gftopk 1.3 (changes as published in TUGboat)
- gftopxl 2.1
- gftype 2.2
- pktopx 2.3 (changes as published in TUGboat)

- pktype 2.2 (changes as published in TUGboat)
- pxtoch 1.1 (with change contributed by William LeFebvre)
- pxtopk 2.3 (changes as published in TUGboat)
- mft 0.3 (a formatting program for METAFONT source files)

Fonts:

- Fonts in `mf` source format. The full Computer Modern as released from Stanford. Utility fonts for character proofs etc. (not made with `cmbase.mf`). L<sup>A</sup>T<sub>E</sub>X and SliT<sub>E</sub>X fonts are also supplied in `mf` source format.
- Fonts in `gf` format. Just about all standard shapes and sizes in 118, 200, 240, and 300 `gf(dpi)` series. The `gf` files for the principal L<sup>A</sup>T<sub>E</sub>X symbol fonts are in a separate list. The Euler fonts are supplied in `gf` format only. (If you want `[euler].mf` source files, you should get in touch with the American Mathematical Society, which will make these available under license.)
- Fonts in `pk` format. AMS fonts—Cyrillic, and special symbols (created with old METAFONT-in-SAIL). All but the `*.590pxl`, `*.1000pxl`, `*.1200pxl` and `*.1500pxl` fonts are in `*.pk` format. The relevant directories include `sh` scripts (e.g. `makepxl300.sh`) for expanding the entire list. See the `*.list` files in each font subdirectory for an idea of the size of the expanded font directory.
- Fonts in `pxl` format. Only the bare minimum set declared in `plain.tex`, in 590 1000 1200 and 1500 sizes, as mentioned above. See also `mcyr`, `msxm`, `msym` (amsfonts).

The distribution has grown to the point of needing a small support organization, which has received a grant from the National Endowment for the Humanities to continue and expand the service of software distribution. Even so, it has become necessary to raise the price slightly. Amateur economists may be amused by the fact that this is one area where volume does not bring lower costs.

**How to order**

To order a full distribution of T<sub>E</sub>X, send \$100.00 (foreign sites \$110.00, to cover the extra postage) payable to the University of Washington to:

Pierre A. MacKay  
Northwest Computer Support Group, DW-10  
University of Washington  
Seattle, Washington 98195

The normal distribution is a tar tape, blocked 20, 1600 bpi. If you need 1/4 inch streamer cartridges for the SUN, be sure to tell us. Although we have had problems previously with cartridge tapes, we can usually hope to get them out quite fast now.

## Typesetting on Personal Computers

**A Bug in T<sub>E</sub>XTURES v0.95 Prerelease**

Gerhard F. Kohlmayr  
Mathmodel Press

**ABSTRACT.** Using a current version of T<sub>E</sub>XTURES™, a discrepancy between Macintosh<sup>1</sup> Plus screen display and ImageWriter<sup>2</sup> printout is documented.

The purpose of this note is to call attention to an implementation error<sup>3</sup> in T<sub>E</sub>XTURES<sup>4</sup> v0.95 prerelease. The control sequences (see [1] and [2]) '\ne', '\neq', '\not=', '\not\equiv', '\not\in', and '\not\subset' are **correctly displayed** on the Mac+ screen, but **incorrectly printed** by the ImageWriter. For example, the *correctly displayed* inequality ' $1 \neq 2$ ' is *incorrectly printed* as ' $1 = 2$ ' and the *correctly displayed* negated membership relation ' $x \notin y$ ' is *incorrectly printed* as ' $x \in y$ '. (These symbols are, no doubt, of great interest to the average mathematician.) No attempt has been

---

I wish to thank P. D. F. Ion for sending me a preliminary version of the A<sub>M</sub>S-T<sub>E</sub>X macro package for the Macintosh Plus and for his assistance over the telephone when I had a problem getting A<sub>M</sub>S-T<sub>E</sub>X running. A<sub>M</sub>S-T<sub>E</sub>X™ American Mathematical Society.

<sup>1</sup> Macintosh is a trademark licensed to Apple Computer, Inc.

<sup>2</sup> ImageWriter is a trademark of Apple Computer, Inc.

<sup>3</sup> Also called simply a bug.

<sup>4</sup> ©1987 Kellerman & Smith •™ Addison-Wesley.

made to find all control sequences which lead to corrupted ImageWriter printouts.

How can one resist the temptation to introduce the term WYSINAWYG (what-you-see-is-*not-always*-what-you-get)?

Barry Smith of Kellerman & Smith has supplied a work-around for this problem, as described in the following note. I am grateful to him for responding within a few days to a first draft of this note by offering, over the telephone, a hands-on demonstration of the ImageWriter bug-fix. I tried it and it worked to my satisfaction.

## BIBLIOGRAPHY

1. Donald E. Knuth. "The T<sub>E</sub>Xbook," American Mathematical Society and Addison-Wesley Publishing Company, Providence, Rhode Island and Reading, Massachusetts, 1984.
2. M. D. Spivak, Ph.D., "The Joy of T<sub>E</sub>X," American Mathematical Society, Providence, Rhode Island, 1986.

**Work-around for an ImageWriter Problem Affecting Output from T<sub>E</sub>XTURES**

Barry Smith

The failure of the \not character to print in output to the ImageWriter is due to the ImageWriter's improper handling of characters having zero width. There are two such characters in the Computer Modern fonts, both in cmsy: the \not character already cited (character "36) and the piece character that forms the left end of the \mapsto symbol (↦, character "37).

The work-around is fairly straightforward, but requires a special piece of software — a font editor: either Resedit or Fontastic will do the job. The change required to make zero-width characters appear is to reset their width to 1 pixel. This is sufficient to avoid the ImageWriter bug, but not enough to cause problems with character positioning or appearance.

If Apple ever gets around to handling zero-width characters in a more intelligent manner, this problem will go away.