Editor's note: This issue is especially large because all items received by press time have been included. The Editor believes that information should be relayed to the TUG membership as soon as possible, so nothing is being held over for another issue.

* * * * * * * * * * *

General Delivery

* * * * * * * * * * *

SITE COORDINATORS
Robert Welland

The following people, who are bringing up TEX on machines at their institutions, have agreed to be site coordinators. Their primary responsibility is to get TEX running and when this is done to write up a report for TUGboat. They have also agreed to answer a limited number of questions; they offer this help for free and the time to do so comes from very busy schedules. If you are not involved with bringing up TEX, please wait for the site reports to appear in TUGboat. Hopefully, they will answer most questions and make manageable the burden the site coordinators will have to bear.

If you are bringing up TEX on one of the following machines please inform the appropriate person; otherwise send the information to the TUGboat editor, Robert Welland.

Machine	*Coordinator*
Burroughs B68000	Scott McCourt Burroughs Corp. B.C.S. Project Corporate Drive, Commerce Park Danbury, CT 06810 203-794-0191 ext 515
CDC Cyber	Thea Hodge University Computing Center 208 Union St. S.E. University of Minnesota Minneapolis MN 55455 612-373-4599
DEC10 running under TOPS-10	Phil Sherrod Box 1577 Station B Vanderbilt University Nashville, TN 37235 615-322-7311 ext 2951
DEC20 running under TOPS-20	Patrick Milligan Bell Northern Research Inc. 685 A Middlefield Rd. Mountain View, CA 94043 415-969-9170 ext 2837
IBM 370	Eagle Berns Polya 207 Stanford Center for Information Processing Stanford University Stanford, CA 94305 415-497-4382
Univac 1100/82	Ralph Stromquist Academic Computing Center University of Wisconsin 1210 W. Dayton St. Madison, WI 53706 608-262-8821
VAX	Monte Nichols Sandia National Laboratories Livermore, CA 94550 415-422-2706

* * * * * * * * * * *

CHAIRMAN'S REPORT
Richard S. Palais

The Steering Committee of TUG met for most of the day on January 9 at the San Francisco Hilton. Much of what transpired at that meeting is reported on elsewhere in this newsletter. I would like to concentrate here on two aspects of the discussions. The first of these is a continuing strong division of opinion on the question of TEX "maintenance", a matter that many will remember already evoked considerable debate at the first TUG meeting. The disagreement is over which of two goals, both obviously desirable, should take precedence. One goal is that TEX should remain as "free" as possible and the other is that TEX should be as carefully and professionally maintained as possible. At one extreme are the large "production" users who would like rapid and dependable advice and help with all their software problems. For them TEX will be one module in a complex system. They have deadlines to meet that require that all these modules work, and they are used to, willing, and able to pay up to several thousand dollars per year to have real or imagined bugs exorcised on the spot and have their software tailored, tuned, and customized for them. At the other extreme is the single, small user with little or no money to spend but competent, willing and able to invest his time and effort in "hacking TEX" for himself. Complicating matters is a third and perhaps over-riding goal, the need to assure that there remains a single, standard "TEX", compatible across many machine architectures and output devices. Fortunately these goals and the constituencies supporting them are not really conflicting, but rather orthogonal. With care and compromise there does not seem to me to be any serious reason why the various categories of TEX users cannot all have their needs met. But it is clear that to avoid nonproductive conflicts and polarizations everyone in TUG will have to keep in mind that the TUG membership is anything but homogeneous and several different options will frequently have to be provided to satisfy all the different classes of TEX users.

The second matter I would like to discuss is the Steering Committee's decision to call for a TEX Implementation Workshop at Stanford in the middle

of May. This will be a two day meeting. One day will be a TEX demonstration day, open to all present and prospective TUG members. This is meant to give an opportunity to become familiar with the various components of the TEX system, and in particular with the different output device options. (At present at Stanford it is possible to have a TEX-produced DVI file output on any one of a Xerox XGP or a Versatec electrostatic printer/plotter, an Alphatype CRS typesetter, or a Canon or Xerox (Dover) laser printer.) The other day is aimed primarily at those actively or prospectively engaged in the implementation of TEX systems and the goal is to maximize the amount of help and information these people can exchange with each other and with the central TEX team at Stanford. The ultimate goal of this implementation project is to be able to supply "off the shelf" to anyone desiring it all the components of a completely working TEX system. Let us consider for the moment what these components are:

(A) TEX-in-Pascal.
 (1) System independent part.
 (2) System dependent part.

(B) Font files.
 (1) Font information files (device independent).
 (2) "Character shape" files.

(C) DVI-to-hardcopy back end.
 (1) Output device hardware interfaces.
 (2) Software for output device interfaces.
 (3) Queuers, spoolers, device drivers for output devices.
 (4) Character shape file "pipeline" from host disk system.

Part (A) (together with B1) is what is necessary to produce DVI output files from a valid TEX input file. Now A1 has long been complete, and A2 is either completed or nearing completion for a wide spectrum of host machines of different manufacture, architecture and operating systems (DEC TOPS-10, TOPS-20, VAX VMS, VAX UNIX; IBM 360/370; CDC Cyber; Univac 1100). I think that we can look forward with some confidence to the May meeting as marking the virtual completion of this first phase of TEX implementation. Now as for part B, the creation of a basic font library, that too is essentially complete. The whole family of CM fonts (and others besides) now exist as **METAFONT** programs. Recall from this column in the first number of the TUG newsletter that **METAFONT** not only creates the device independent font information file (containing the size, spacing, kerning, and ligature information needed by TEX to create a DVI file), but also, once a simple interface program is written for a given output device, **METAFONT** will create the character shape files, in the form of raster patterns stored as, say, matrices of zeros and ones. Such **METAFONT** interfaces have now been written for over a half dozen output devices, running from the super high resolution (5300 dot/inch) Alphatype CRS to the low resolution (128 dot/inch) Florida Data impact printer.

So what is rapidly approaching is the final phase of the TEX implementation program, the creation of the back-end systems which for a given host mainframe and output device will, from the DVI file and character shape files, produce the hardcopy output. Now as David Fuchs has remarked, life would be quite easy if output devices had built into them enough disk-type storage to handle all the character shape files for sixty-four fonts of 128 characters each and enough logic to process the DVI files into raster scan lines. One would still have the (rather trivial) job of writing for each operating system spoolers and queuing programs to send DVI files over a serial line to this ideal output device in an orderly fashion, but one would be able to avoid a host of other small headaches that real world output devices force one to deal with. Since, in fact, output devices usually have no usable general purpose microcomputer built in, one must interface the host computer to the output device via a microcomputer able to speak to both. Also, since all the character shape files that must be accessible to process a complex DVI file can in principle run in the megabyte range, economic considerations mandate that with current technology these files must be kept on the host computer disk memory, and then downloaded as necessary to a small floppy disk system associated to the microcomputer interface over the (for simplicity) serial line joining it to the host computer. Now, designing the hardware interface from off the shelf items and writing the software to make it all go is not a major project for an expert systems programmer who understands the format of DVI files and character shape files and knows how to communicate scan lines to the output device. Perhaps a month or two of hard work will suffice. What is frustrating is that these systems are extremely sensitive to small differences in the various protocols of operating systems and output devices, so if there are m of the former and n of the latter one could easily end up doing this same work mn times. Of course common sense tells us that good planning should be able to reduce this to more like $m + n$ times. (For example, David Fuchs could probably interface one more output device to TOPS-20 in under a week.)

A major reason for the May meeting is to reduce as much as possible unnecessary duplication of effort in this final part of the program to make TEX generally available.

* * * * * * * * * * * *

REPORT ON THE TUG STEERING COMMITTEE MEETING

The TUG Steering Committee and several observers met at the San Francisco Hilton on January 9, 1980. Below is reported the gist of that meeting as recovered from tape recordings and my notes. Since several topics re-emerged throughout the meeting, I have not reported in any order related to that of the meeting.

Robert Morris

Attendance

The following attended:

Barbara Beeton, AMS Providence
Max Díaz, Stanford
Barry Doherty, AMS Providence
David Fuchs, Stanford
Ellen Heiser, AMS Providence
Don Knuth, Stanford
Leslie Lamport, SRI
William LeVeque, AMS Providence
Patrick Milligan, Bell Northern Research
Robert Morris, UMASS/Boston
Evon Motiska, Stanford
Monte Nichols, Sandia Labs
Richard Palais, Brandeis
Lynne Price, Bell Northern Research
David Rogers, University of Michigan
J. L. Selfridge, Math Reviews
Phil Sherrod, Vanderbilt
Michael Spivak, Decatur, Ga.
Rilla Thedford, Math Reviews
Luis Trabb-Pardo, Stanford
Bob Welland, Northwestern
Sam Whidden, AMS Providence
W. B. Woolf, Math Reviews
Ignacio Zabala, Stanford

Treasurer's Report

Sam Whidden gave the treasurer's report, attached. The cost of producing the first newsletter exceeded the treasury by $419. It is estimated that an additional $3600 is needed for two issues in 1981.

Miscellany

Pat Milligan and Lynn Price of Bell Northern Research have been extensively developing macros in-house and have had successes making overhead slides, Hebrew, and special graphics.

AMS-TEX

Mike Spivak reported that version —1 works as indicated in *The Joy of* TEX, where not-yet-implemented features are indicated in handwritten marginalia. The *Joy* was *not* processed by *AMS*-TEX, which will now enter its field testing phase as people begin to use it. An order form for a tape is included with the manual, for sale at AMS headquarters. The first finder of each manual misprint will receive a $1 bounty, and the first finder of each *AMS*-TEX bug a $5 bounty. Bounty may be claimed by writing Mike at the address in *The Joy of* TEX. The *Joy* was produced on the Providence Alphatype, which is exhibiting some backlash problems resulting in distortion of some vertical lines.

AMS-TEX has too many macros for easy use in the SAIL TEX running now at Providence. The Pascal version is not expected to have the size limitations which caused the problems, which in any case can be changed by recompiling TEX with bigger values of hashsize. The Pascal version will provide 3 bits more address space for internal memory than the SAIL version and such problems will not be serious. A similar problem with memsize appears when setting multi-column output. These should also disappear in the Pascal version. Making *AMS*-TEX macros more efficient will help, which Mike will do this week.

Maintenance

The administrative burden of maintaining TEX has become too large for Stanford to support on the informal basis they do. Throughout the meeting at various times debate raged on the appropriate mechanism for maintenance. Since this is inextricable from membership fees, a Finance Committee was formed to recommend a maintenance policy, to recommend a membership fee policy, and to explore sources of support, e.g. foundations. This consists of Sam Whidden, Luis Trabb-Pardo, chairman, Bob Morris, Pat Milligan and Monte Nichols. It seemed that everyone agreed that TUG would run a TEX switchboard whereby someone would be paid to tell callers who can answer their questions. Luis and the Stanford people are spending too much time doing this and answering the questions. The switchboard could also refer people to an up-to-date list of consultants for hire.

The (unsettled) argument about maintenance varied between two positions: (a) Some organization with either an explicit financial interest or an in-house TEX support facility should maintain TEX at TUG's expense. (b) There should be no financial burden whatsoever on the membership and no particular TEX maintenance should be endorsed by the

Users Group (see separate articles in this newsletter). Knuth's intention is that the released Pascal TEX will be a single stable core (aside from the system dependent module) which can be uniformly maintained for all versions and should not have any supported enhancements.

TUG Meeting

No general TUG meeting will be called until the Pascal TEX is released. However, an Implementor's Workshop has been called for May 14–15 at Stanford. This will be coupled with a TEX open house comprising demonstrations for people who don't know what TEX is. Details are elsewhere in this issue.

Pascal TEX

A detailed report of each architecture appears elsewhere in this issue. Knuth expects to read the Pascal version's code this spring very carefully before its public release. When released, it will be frozen with no enhancements or changes perhaps aside from bug fixes. At present, the only fully operational Pascal version is the TOPS-20 implementation. One problem is that implementors have also to get their output hardware working in order to see output. In general interfacing output devices is proving a greater share of the implementation efforts than people imagined, but has nothing directly to do with bringing up Pascal versions.

Education

Knuth is making a video tape to teach TEX to users. He intends for this tape ultimately to be available through TUG.

Output Devices

Phil Sherrod: It took a week to get the SAIL version working on TOPS-10, but nearly a year to get all the output device interfaces conveniently working (e.g. spoolers) although only a month to get something up. Finding the right hardware interfacing was lengthy and mysterious. TUG should maintain descriptions of what kind of hardware to buy and what the software interfaces involve.

Phil will write an article describing the tribulations of output device implementors.

There was a belief expressed that the output device vendors would have to be involved in output interfaces. Cooperation greater than that already provided by manufacturers of existing devices will be needed for wide applicability. Luis Trabb-Pardo expressed the belief that more intelligence needs to be provided in devices in order to relieve burden on the host, which will allow less system dependent software. The problem is that Xerographic printers are or soon will be selling for about $3000 for the printing engine. The interface will cost about the same, bringing the OEM cost to about $6–7,000. The end-user prices will be around $20,000 for complete printing systems. These will have sufficient intelligence to take DVI files more or less directly. A similar arrangement based on electrostatic printers should sell for around $10,000 end-user system price.

Math. Reviews has interfaced a Florida Data dot matrix printer with the same mechanism (a $3500 one-board Z80 system) to drive it in graphics mode. The device has 128 dots/inch and might be suitable for very rough copy. It is doing TEX output at about 30 seconds per page, which is a little slower than the electrostatic devices.

Varian, Versatec, Dover, and Alphatypes are working at several sites. The Dover does not accept DVI files and is in any case not commercially available.

* * * * * * * * * * * *

1980 TUG TREASURER'S REPORT

Samuel B. Whidden

During 1980 the first issue of TUGboat appeared. The costs associated with its printing and distribution amounted to $1,719. (Not included in this figure are costs for services provided by AMS professional staff.) As of December 31, 1980, 130 membership applications have been received for for a total income of $1,300.

Income:	Membership		$1,300
Expenses:	Printing	$1,232	
	Postage	371	
	Mailing/Labor	116	1,719
Balance (as of 12/31/80)			($ 419)

Based on the costs for 1980, it is anticipated that direct costs associated with the production and distribution of two issues of TUGboat during 1981 will be approximately $3,600, with the AMS continuing to contribute the services of its professional staff.

Respectfully submitted,
Samuel B. Whidden, Treasurer
January 5, 1981

(*Note*: As of 2/6/81 a total of 258 paid membership applications had been received.)

* * * * * * * * * * * *

INFORMAL TUG SESSION

Robert A. Morris

On the afternoon of January 9, various TUG Steering Committee members and other interested

people met at the Hilton for informal discussion. Among the people I can remember were there were: Lynne Price and Pat Milligan (Bell Northern Research), Leslie Lamport (SRI), Luis Trabb-Pardo (Stanford), Rilla Thedford (Math Reviews), Arnie Pizer (Rochester). Possibly I have missed some. I have reported below some of the wishes, rumors, and reports from this meeting and other sources. Nothing is guaranteed accurate!

Lynne and Pat have extensive macro experience and have made slides, Hebrew, and are advocating TEX as a standard for in-house technical documentation in their organisation.

Some desires: improved user interfaces. Lynne will start collecting complaints and suggestions. $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX is an example.

Many people want a TEX preprocessor which can run on small machines with which people can test the syntax of their TEX input. (But Unidot has its C version running on an Onyx UNIX system with a Versatec printer and hopes for release soon. This could presumably be used even without an output device. It apparently is based on the SAIL version of TEX. Will it be released with all the changes which end up in the Pascal version? It will be for sale.)

Luis: In principle, all that is needed for the use of an arbitrary printing engine is cooperation from the vendor in providing (a) Font Metrics for each font; (b) If the galley proof cost is high, will they provide proof mode encodings (e.g. at 200 dots/inch), not only thereby protecting their own font investment but allowing users to run proof mode versions of their fonts on a proof device? (c) Is the typesetter language available to people to write DVI-to-device drivers? (d) Is the manufacturer willing to include math fonts made by **METAFONT**?

A trick if your macro packages are too big to fit (which shouldn't happen very much in the Pascal versions): redefine as null macros which will not be used again. This will return the space to the memory manager.

Leslie Lamport: A trick to avoid un-matched brace syntax errors: When using a screen editor like EMACS, use a macro which creates matching braces with nothing between them except the cursor. The closing brace is then automatically there after the text entry is finished.

* * * * * * * * * * * *

Editor's note: The following two articles give divergent views on the subject of how the TEX program is to be maintained in the future. Readers are invited to comment, and to make known their own views on the subject before the next meeting of the TUG Steering Committee in May.

* * * * * * * * * * * *

A POSITION ON TEX MAINTENANCE

Robert Morris, UMASS/Boston

There are at least two diametrically opposed maintenance/distribution models we can consider. The production user, like AMS or a commercial user, wants something like a fully supported TEX. Such a user has a calculable, or at least identifiable, financial penalty which it incurs when the software it is using does not work. On the other hand is the university user facing little or no budget resources which it can devote to buying support, but also having no particular time constraints and having a pool of talent—its students—which can tweak non-working software. Since I am in the second community, I would like to argue in favor of TUG's involvement being closer to the second model, while facilitating the other class of users solving their support needs at their own expense.

I believe that few universities would benefit by paying $1,000/year membership fee to TUG. Indeed, $200 seems too much to me for an organization which can wait weeks to get sick software fixed. Instead of contributing to extensive TEX support, I suggest that there be designated distribution sites for each architecture, selected from some organization heavily using TEX on that architecture. These sites would make standard release tapes at cost and would incorporate bug fixes at designated intervals (quarterly?). Their interest would lie principally in being the funnel for proposed bug fixes (which would often be proposed by the discoverer of the problem) and thereby having the first and widest perspective on maintenance. Presumably these would be organizations which are already maintaining TEX in-house and thus have sufficient expertise to recognize whether a bug report is in fact a TEX problem or a user problem.

The other side of this essentially un-supported TEX is that users can tinker with TEX and circulate their own "enhancements". On the one hand, this is contrary to desires that Knuth has expressed. On the other hand, it is bound to happen when sources are distributed, and I am not convinced it is bad. The most successful model of this kind of unsupported source distribution is the UNIX operating system. UNIX is distributed free to educational users with source licenses. Tapes are made at cost by the licensor, Bell Laboratories. Often, these

releases do not work on the precise configuration the licensee has and varying degrees of work are required to bring up the system. Alternatively, users often get copies of the system not from Bell Laboratories but from another site with the same or similar configurations. Many users modify their systems and/or install major modifications made by other sites. This process continued for 6–7 years throughout the life of "version 6 UNIX", the first version in wide circulation. All this experimentation led to two things: a bizarre proliferation of somewhat incompatible versions of UNIX and a substantial base of expertise about the system coupled with a great deal of experimentation toward modernizing the operating system. The result of the former was that UNIX came to be regarded as needing substantial systems programming expertise to keep it running (a false belief which did not take into account the simplicity of the operating system and the ease of dealing with code written in a high-level language). The result of the latter was that version 7 UNIX and that for the VAX have incorporated the results of these experiments and apparently produced a very contemporary and useful operating system which internally looks little like version 6, but to users is very similar. After all this tinkering, the resulting product seems to be useful not only in universities, but at high prices in commercial environments.

On the one hand, such a model seems incompatible with Knuth's position that TEX will be released in such a way that tinkering is un-needed. He prefers that people think of TEX as something which will not need enhancement, but rather will be the foundation of similar future developments which are not TEX but (hopefully) something better. On the other hand, it suggests that the way to find what the something better might be is actually by the kind of refinement which took place in UNIX with wide circulation (among licensees in that case, but presumably among everyone in the TEX case).

It strikes me that any form of TEX support will have its cost underwritten either by an organization seeking to profit from it or by the Users Group seeking to keep to Knuth's idea of a single uniform TEX. My feeling is that the cost, especially to academic users, of the latter is prohibitive (one estimate mentioned at the TUG Steering Committee meeting was $25,000–50,000/year total distributed among 50–100 institutional members).

My guess is that the costs of making fixes to *all* releases, i.e. all architectures, could climb above this just because most sites will not have expertise in all systems.

I am afraid that any commercial organization which assumed "official" responsibility for repairing TEX would insist on reasonable assurances that no one would compete with them, for example that I would not give away bug fixes the way UNIX sites do. Since no such assurance is possible because the software is in the public domain, it seems that the alternative is to have the entire TUG membership pay for the support by membership fees.

I would propose that as a group TUG provide no services other than the dissemination of information about TEX, including the Pascal release. Any bug fixes would be reported but not endorsed by TUG perhaps except at stated intervals when new releases would incorporate them. Production users of TEX would be entirely on their own in finding support at the level they need. I am inclined to argue that there should be several distribution sites, one for each architecture and that they should *not* be sites which have a commercial interest in selling supported TEX. They should be reimbursed for their direct expenses by each recipient in the form of a nominal ($50–200) fee for providing the release tape and documents, and some attempt should be made to ascertain a reasonable level at which TUG will annually reimburse them for less tangible related expenses, e.g. time spent consulting with people having difficulty installing the release, time spent evaluating bug reports, etc. The balance of TUG's budget should be spent for the "switchboard", the newsletter, and the expense surrounding incorporating bug fixes at the stated (infrequent) intervals.

Ordinary TUG membership meetings should be financed largely out of meeting fees which should be appreciably smaller for educational users than for production users.

There will be two classes of institutional memberships: educational users and production users. An educational user is a non-profit educational site which is using TEX only for instruction or for the production of publicly accessible research or instructional documents. A production site is a site which is using TEX principally for the production of administrative, clerical, or commercial or published documents. A production site is using TEX because it expects to save or make money by doing so, whether directly or indirectly. Such sites include not only commercial enterprises, but also the AMS, the publications departments of universities, and the in-house document preparation centers of not-for-profit research organizations.

* * * * * * * * * * * *