OFFICIAL ANNOUNCEMENTS

1982 Membership Dues

1982 dues for individual members of TUG will be $15. Membership privileges will include all issues of TUGboat published during the membership (calendar) year. All new members and other persons inquiring about TUG will be sent TUGboat Vol. 1, No. 1, but 1981 issues will be sent only to persons paying the 1981 dues of $10. Beginning in 1982, foreign members will be able, on payment of a supplementary fee of $12 per subscription, to have TUGboat air mailed to them.

TUGboat Schedule

Volumes of TUGboat are numbered on a calendar year basis. Volume 1 appeared in 1980, Volume 2 corresponds to 1981, and 1982 will bring Volume 3. Volume 1 consisted only of issue No. 1, dated October. Three issues are planned for Volume 2: No. 1 appeared in February, and No. 3 is planned for November. No schedule has been determined yet for 1982.

The deadline for submitting items for Vol. 2, No. 3, is October 1, 1981. Contributions on magnetic tape or in manuscript form are encouraged; editorial addresses are given at the bottom of page 2, and a form containing instructions for submitting items on tape is bound into the back of this issue.

It has been necessary to reprint back issues of TUGboat to fulfill the requirements of the growing membership. Each member is entitled to receive all issues which appear during his membership year, as well as Vol. 1, No. 1. If you have not received any issue to which you are entitled, instructions for obtaining such issues are included on the form referred to above.

* * * * * * * * * * *

General Delivery

* * * * * * * * * * *

EDITOR'S REMARKS

Robert Welland

We thank Lynne Price for taking on the responsibility of editing our Macro column; it is a complex task and we are thankful that it is in such talented hands. In the future, please submit all macros to

Lynne A. Price
CALMA
Research and Development
212 Gibralter Drive
Sunnyvale, CA 94086

We also thank Barry Smith of Oregon Software for getting TeX up and running on the VAX (see the VAX/VMS site report, page 34) and for making it easily available to all VAX users. Because of this work, we will see TeX flourish at very many sites.

Due to the hard work of Thea Hodge and Michael Frisch of the University of Minnesota (see their site report on page 28), we hope to see TeX up and running on Cyber machines sometime this fall; may the North Star guide them to success.

Lastly we extend the membership's gratitude to Barbara Beeton and Sam Whidden of the AMS whose hard work has made the TUGboat newsletters possible.

* * * * * * * * * * *

Editor's note: The TUG Chairman, Richard Palais, is on leave for a year. At the Steering Committee meeting in May, Michael Spivak was appointed to serve as temporary Chairman until Dick's return.

* * * * * * * * * * *

CHAIRMAN'S REPORT

Michael Spivak

Since I am substituting for Dick Palais as Chairman of the TUG Steering Committee during the next year, I suppose that I ought to emerge briefly from the dimness of the AMS-TEX macro engine room and report on the view from the bridge.

Up here it's all inchoate brightness—everything's presently in a fog, though there's the promise of smooth sailing ahead. By the time of the Cincinnati meeting in January, the official Pascal TEX should be published, and more important, up and running at many more sites. If you have encountered and solved any particular problems bringing TEX up, your experiences will undoubtedly be of interest to others who want to implement TEX on the same, or similar, systems. If possible, please present your installation and/or use experiences at a session of the Cincinnati meeting; see the preliminary announcement by Tom Pierce on page 8. Perhaps we'll soon be able to stop worrying about getting TEX running, and can concentrate on using TEX. Two fundamentally opposed philosophies of how TEX should be supported were spelled out by Bob Morris and Sam Whidden in the last issue of TUGboat, and it will certainly be interesting to find out just *how* much support is going to be needed, since this will obviously influence the final decision. Actually, it seems that the problem of getting TEX running (i.e., producing dvi files) will be much easier to solve than the problem of getting the files printed, because of the variety of printers used and the secrecy about their inner workings. Perhaps this should be the next major problem that TUG could make a systematic attack on.

Of course, TEX is already up and being extensively used at some places, and more and more macro packages are being produced to get TEX to do just about everything except shine your shoes and write the papers for you. At the present stage, there are clearly still many tricks to be learned (as Don said, we are just beginning to scratch the tip of the iceberg). Even if a macro package performs some function that isn't of particular importance to another macro writer, it may contain some tricks that will be useful. Perhaps we should encourage more people to send in special tricks, or emphasize such tricks in their macro packages; eventually a "standard library" of tricks could be compiled. (Hours of pestering Don have produced some basic tricks, documented in the article "Macro Madness" (see page 50), that may help people to make TEX's macro facility work more like the 'programming language' that many have wished for.)

As this last paragraph has indicated, my own particular interest in using TEX is to get it to typeset anything a mathematician would want with minimal understanding on the part of the typist. Obviously the interests of other TEX users and implementors are going to be quite different. One of the problems with our last meeting was its undifferentiated nature. Although almost everyone got quite a bit out of some particular talk or meeting, it wasn't easy to know beforehand which one it would be. This is probably only to be expected at the initial stages, especially since so many different levels of TEXpertise are being addressed, but with Tom's help the Cincinnati meeting ought to be better structured, so that people can know what will be useful to them, and what can be skipped. Perhaps we'll even be so organized that we can propose the organization for the next meeting. Let's hope so!

* * * * * * * * * * *

REPORT ON THE TUG STEERING COMMITTEE MEETING

Robert Morris

The TUG Steering Committee meeting took place in two sessions. The first, on May 13, simply set the loose agenda for the second, which was a public meeting on the evening of May 14.

The following actions were taken (a few of these may have been taken at the loosely organized general membership meeting on May 15):

a. By acclamation Mike Spivak was declared Chair of the Steering Committee. Richard Palais will be out of the country for a year.

b. Personal dues will be raised to $15 for 1982, but no institutional dues are contemplated pending TUG offering something to its members beyond the newsletter.

c. The Treasurer's report was approved; a version updated through June 30 appears on page 5. In summary, the individual membership fees and excess workshop revenue will cover the publication of TUGboat and minor administrative expenses for this year.

d. The idea of having architecture specific implementors' workshops, preferably at a successful site, was endorsed. These would be highly technical and financially self-supporting. Vanderbilt may organize one for TENEX sites; see page 28 in this issue for an announcement.

e. A tape standards committee was established to propose formats for the exchange of TEX files. A first proposal is put forth by Patrick Milligan on page 10.

f. Lynne Price agreed to edit the Macros and Problems columns in TUGboat, and to serve as focal point for discussion on the next generation of TeX, with emphasis on user-friendliness.

g. It was agreed to call a general membership meeting to coincide approximately with the winter meeting of the American Mathematical Society next January in Cincinnati; see the preliminary announcement by Tom Pierce on page 8.

h. Don Knuth announced his desire/intention to have TeX fully frozen by the end of the year, and to publish the theory and workings of TeX early in 1982.

i. It was decided that the architecture coordinators should not in general be those actually implementing, in order to shield the implementors from repetitive questions. This has worked well for the VAX/VMS implementation and will be gradually accomplished for the other architectures. Site coordinators are listed on the inside front cover, and their addresses are given on page 2. If you have/want current information, please contact them.

Minutes respectfully submitted,
Robert A. Morris
Secretary

Editor's note: Attendees at the meeting may submit additions and corrections to the minutes in writing to the Secretary.

* * * * * * * * * * *

TUG TREASURER'S REPORT

June 30, 1981

Beginning balance, January 1, 1981:			$(419)
Income: Membership[1]		$1,555	
Tape leasing		400	
Workshop[2]		7,445	9,400
Expenses[3]:			
TUGboat Vol. 2, No. 1: 500 copies			
Printing	$1,012		
Postage	320		
Clerical labor	60	$ 1,392	
Reprinting TUGboat:			
Vol. 1, No. 1: 300 copies		195	
Vol. 2, No. 1: 300 copies		655	
Steering Committee luncheon, San Francisco Jan. 81		170	
Workshop[2] expenses		236	(2,648)
Estimate of future 1981 income:			
Membership (100 members)		$ 1,000	
TeX tape sales/leasing		1,000	2,000
Estimate of future 1981 expenses:			
TUGboat Vol. 2, Nos. 2&3: 800 copies			
Printing	$3,200		
Postage	900		
Clerical labor	200	$ 4,300	
Reserve for 1981 expenses for Cincinnati meeting, January 1982		1,000	
Support for Stanford TeX Coordinator[4]		3,600	(8,900)
Subtotal:			$(567)
Anticipated receipts in 1981 against 1982 individual membership (50% of membership)			4,500
Balance (estimate to December 31, 1981)			$ 3,933

Notes:

1. Total membership is 495, of which 30 are complimentary; of these, 371 members are domestic and 124 foreign.
2. The Implementors' Workshop held at Stanford, May 14-15, 1981, was attended by 92 participants.
3. Not included in these figures are costs for services provided by AMS professional staff, including programming, reviewing and editing, answering telephone inquiries, maintaining the mailing list, and other clerical services.
4. Professor Arthur Samuel is acting for Luis Trabb-Pardo as TeX coordinator, answering questions, distributing tapes, and fixing bugs in the TeX source code. Luis has asked, and the finance committee has agreed, that TUG contribute to Professor Samuel's support.

Respectfully submitted,
Samuel B. Whidden, Treasurer

* * * * * * * * * * *

PROPOSAL FOR INSTITUTIONAL SUPPORT OF TUG

Robert Morris

Late last week (June 14) the Finance Committee met with an unusual opportunity to fund something which has made me change my previous position about TUG institutional membership. Barring an obstruction due to an Air Traffic Controllers strike, we are sending the chairman of TUG, Mike Spivak, to the ANSI standards committee on Text Processing Languages, X3J6.

This committee will be considering a number of possibilities for the processing of mathematical text, and one of the Steering Committee members, Lynne

Price, will be attending as a member. However, Lynne can not attend the beginning of the meeting and felt it important that TeX be represented at least informally by someone knowledgeable.

Acting in hastily convened and loosely organized telephone meetings, we agreed to pay the cost of Mike's attendance at this meeting as our observer, even though TUG has no funds in its budget beyond those needed to pay for the newsletter. Approximately $1000 will be borrowed from the AMS to be reimbursed from future TUG income.

In the Steering Committee meeting (see my minutes, page 4) it was agreed that we would propose no institutional dues until we had some proposal for use of such money to the benefit of the membership. Here is such a benefit: representation of the TeX user community at standards committees and other organizations which may be in a position to influence the use or restriction of text processing systems (for example, I could envision also presentations to governmental agencies who might be promulgating standards for government documents).

Another benefit I think should accrue to paying institutional members is an annual (?) tape of contributed macros and (perhaps) a copy of AMS-TeX when it is in its "positive versions" (in the current pre-release versions I am enthusiastic about distributing it at cost to anyone who wants to test it. Later, I would make it a benefit of institutional membership).

Thus I now argue for the following dues structure:

Individuals	approximately the cost of TUGboat
Educational institutions	$100
Non profit institutions using TeX in house	$250
All others	$500

Note that I have included all commercial organizations and all users of TEX who use it to produce publications for sale (e.g. the AMS and university presses) as one class of users.

I hope the precise figures and the ratios will be the subject of much discussion in this forum, because I will ask for formal ratification of some such structure at the annual meeting in January.

I will collate any replies this note brings. Please mail them to me at:

- (before Sept 1): IBM Cambridge Scientific Center, 545 Technology Square, Cambridge, MA 02139.
- (after Sept 1): Dept. of Mathematical Sciences, UMASS/Boston, Boston, MA 02125.
- Arpanet address: ram@mit-mc.

If you are especially anxious that the full text of your reply be published in TUGboat, please so indicate.

Editor's note: The X3J6 meeting described above has been rescheduled, and Lynne Price will probably attend rather than Mike Spivak, so that TUG funds will very likely not be required. Bob's new position, in favor of institutional support, is not affected by this change, a fact he has confirmed in a telephone conversation.

* * * * * * * * * * *

REPORT ON THE TeX IMPLEMENTORS' WORKSHOP, STANFORD, 14–15 MAY 1981

Barry C. W. Doherty

At the TeX Implementors' Workshop in May, 92 people were registered (a complete list follows). The goal was to draw together both those knowledgeable about TeX and those in various stages of the implementating TeX-in-Pascal, from having an interest to having completed the installation, so that there could be communication of the problems and solutions involved.

The first day consisted of a series of planned talks on various aspects of TeX, from advanced usage to desirable features of Pascal compilers and technical details of TeX's output. On the second day, a series of informal sessions focused on people's principal interests and concerns, attempting to provide the information most necessary for those trying to install TeX and to gather the major unsolved problems hindering such installation.

Some of the more 'formal' talks either appear as articles in this issue of TUGboat or will appear in subsequent issues. Similarly, a number of the topics addressed during the second day have generated communications that appear here. The range of interests was large, with the result that many participants felt that much more could (or should) have been said about each of the topics. (Perhaps these communications will stir such a discussion in these pages!)

The schedule

First day (May 14*th*)

9:00–10:00 **Donald Knuth** *"TeX debugging aids"* A detailed analysis of sample TeX input using information available through features built into TeX (such as \trace and \ddt). It is hoped that a presentation of this talk will be available for the next issue of TUGboat.

10:00–10:45 **Ignacio Zabala** *"Pascal-related issues"* Concentration on the characteristics and suitability of various popular Pascal compilers, with suggestions on what to look for in a compiler and how to cope with the compiler one has. (*See this issue, p. 16.*)

11:15–12:00 **Ignacio Zabala** *"The system dependent module of TEX-in-Pascal"* The Pascal elements of TEX and their implications. (*See the articles by Lawson, Zabala and Díaz,* TUGboat Vol. 2, No. 2, pp. 20, 32.)

1:15–2:00 **David Fuchs** *"Different output formats, conversion issues"* Largely a discussion of TEX's DVI file format. (*See this issue, p. 12.*)

2:00–2:45 **Luis Trabb-Pardo** *"From DVI to paper"* General discussion of translator (driver) programs (from DVI to something a specific device understands), and the role of spoolers/servers in scheduling and queueing—features and characteristics, downloading of fonts, memory requirements, efficiency.

3:00–3:30 **Frank Liang** *"Hyphenation in TEX"* Discussion of the algorithm used in TEX and comparison with other widely used algorithms. (*See this issue, p. 19.*)

3:30–4:00 **Michael Plass** *"Lines, paragraphs, pages"* Discussion of how TEX functions in this context. See report by Donald E. Knuth and Michael F. Plass, *Breaking paragraphs into lines*, Stanford CSD report CSD-CS-80-828.

A panel discussion had been scheduled to begin at 4:00; talks were running longer than planned as a result of discussions following most. Instead Don Knuth spent a few minutes discussing his plans for TEX, which include a series of three books providing complete documentation on the system (dates are projected completion dates):

- TEX—an entire listing of the Pascal source code, a 'final' user manual, and a history of debugging TEX. (Winter 1982)
- Computer Modern Roman—a description of this font family. (Spring 1982)
- **METAFONT**—similar to the book on TEX. (Winter 1983)

. . . .

Second day (May 15*th*)

9:00–10:00 *"TEX distribution and installation"* General problems of obtaining TEX and of the transportability of both TEX and TEX-related files. Questions were raised about the real utility of the current means of distributing TEX-in-Pascal as two quasi-independent documents (Pascal source code and internal documentation), both produced from the same meta-language source; general opinion seemed to favor distribution of the original source together with the programs (currently implemented only in SAIL) for producing the pieces, to allow each site to tailor the results to its (and its compiler's) needs more easily. One result was the formation of a tape standards committee. (*See the article by Milligan on this committee, p. 10.*)

10:00–11:00 *"***METAFONT** *and fonts"* Interest in both **METAFONT** and in the distribution of fonts. Again, one result was the formation of a committee to look into the problems. (*See the article by Doherty, p. 34.*)

11:00–12:00 *"Son of TEX"* Even before TEX's final release there have been numerous suggestions for what TEX might (or ought to) do. The spirit of these modifications is to allow more specialized typesetting to be done without damaging the compatibility with standard TEX. Some desired features include a more "suitable" input language, more tractable error messages, incorporation of graphics output, non-English hyphenation capabilities, batch mode (rather than interactive processing), and real-time interactive TEX. (*See the article by Price, p. 58.*)

1:00–2:00 *"Macro packages"* Already several major macro packages have been developed (see the documentation on the macro packages by Keller and Díaz, for instance, as well as Spivak's AMS-TEX, in various issues of TUGboat). Here there was an attempt to focus on standards and conventions of possible interest to macro writers: questions of compatibility, consistency in font-naming, conventions for replacing characters found on the Stanford nonstandard terminal keyboards. (*See the articles by Milligan (p. 44) and Price (p. 43) in this issue.*)

2:00–3:00 *"Output devices and their interfaces"* A somewhat more specific examination of some of the more common output devices, their characteristics and what is required of their interfaces.

3:00–4:00 *"Architecture sessions"* About a half-dozen groups formed to discuss their particular problems. Major sessions included IBM, VAX, DEC 10s and 20s, CDCs.

4:00–5:30 *"Output device demonstrations"* This was devoted to Trabb-Pardo's presentation of the Canon Laser Printer (*see his article in this issue, p. 26*) and a tour of BNR given by Milligan (equipment including a Versatec, PERQ, and Alphatype).

* * * * * * * * * * *

Attendees, TeX Implementors' Workshop Stanford, May 14–15, 1981

Adamo, Vincent - Texas A & M University
Amabile, Carolyn - National Information Systems
Ash, William - Stanford Linear Accelerator Center
Ball, George - Washington State University
Beebe, Nelson - University of Utah
Beeton, Barbara - American Mathematical Society
Bennison, John - Brown University
Berns, Eagle - Stanford University
Blair, John - CALMA
Broadwell, Peter - Univ. of California, Santa Cruz
Brown, Malcolm - Stanford University
Buckle, Normand - University of Montreal
Bupara, Sarge - Exxon Office Systems
Carnes, Lance - Gentry, Incorporated
Chaffee, Roger - Stanford Linear Accelerator Center
Conley, Marsha - University of Illinois
Copeland, John
Cralle, Robert - Lawrence Livermore Lab
Dailey, William H. - Letterman Army Institute
Day, Christopher - Lawrence Berkeley Lab
Díaz, Max - Stanford University
Doherty, Barry - American Mathematical Society
Doob, Michael - University of Manitoba
Durling, Bob - University of California, Santa Cruz
Faul, Don - Lawrence Livermore Lab
Faulkner, Thomas - Washington State University
Forster, Doug - Stanford University
Frisch, Michael - University of Minnesota
Fuchs, David - Stanford University
Gittelsohn, Michael - San Francisco State University
Goldby, Alan - University of California, Santa Cruz
Grosso, Paul - University of Michigan
Guenther, Dean - Washington State University
Hickey, Thomas - OCLC, Incorporated
Hodge, Thea - University of Minnesota
Jackson, Calvin - California Institute of Technology
Katagiri, Grace - University of California, Berkeley
Kelley, Al - University of California, Santa Cruz
Knuth, Donald - Stanford University
Lanford, Oscar - University of California, Berkeley
Lindsey, Clark - University of California, Riverside
Mapes, Jeff - Stanford University
Melen, Randy - Stanford University
Milligan, Patrick - BNR, Incorporated
Morris, Bob - University of Massachusetts, Boston
Nichols, Monte - Sandia Labs
Norstad, John - Northwestern University
Nussbaum, Frank - Newline Graphics
Palais, Richard - Brandeis University
Payne, Thomas - University of California, Riverside
Pierce, Thomas - EG&G, WASC, Incorporated
Plass, Michael - Stanford University
Plass, Susan - Stanford University
Platt, Craig - University of Manitoba
Price, Lynne - BNR
von Raesfeld, Mary - National Information Systems
Reier, Warren - Gentry, Incorporated
Renz, Peter - W. H. Freeman and Company
des Rivières, Jim - Carleton University
Robb, Richard - Cemrel, Incorporated
Rosenschein, Jeffrey S. - Stanford University
Ross, Kenneth - University of Oregon
Rushworth, Tom - Block Brothers Industries
Sachs, Jonathan - independent contractor
Samuel, Arthur - Stanford University
Schechtman, Marty - Newline Graphics
Scott, Eric P. - California Institute of Technology
Sears, Chris - San Francisco State University
Sherrod, Phil - Vanderbilt University
Smith, Barry - Oregon Software
Spivak, Mike
Stovall, John - Wycliffe Bible Translations
Stromquist, Ralph - Univ. of Wisconsin-Madison
Tal, Avi - Electis Engineering Incorporated
Thedford, Rilla - Mathematical Reviews
Trabb-Pardo, Luis - Stanford University
Truax, Terry - Mathematical Reviews
Tuttle, Joey - I. P. Sharp Associates
Van Dalen, Gordon - University of California, Riverside
Van den Bosch, Peter - Univ. of British Columbia
Wakabayashi, Nobuo - Stanford University
Weening, Joe - Stanford University
Welland, Robert - Northwestern University
Wheeler, Norman
Whidden, Samuel - American Mathematical Society
Whipple, Edgar - Lawrence Berkeley Lab
Whitney, Lynn - Univ. of California, Santa Cruz
Whitney, Ron - American Mathematical Society
Wilmott, Sam - Block Brothers Industries
Wiser, David - Stanford Linear Accelerator Center
Wolf, Joe - University of California, Berkeley
Zabala, Ignacio - Stanford University

* * * * * * * * * * *

PRELIMINARY ANNOUNCEMENT: TUG MEETING, CINCINNATI, JANUARY 1982

The next TUG meeting will be held in Cincinnati, Ohio, at the Stouffer's Cincinnati Towers from January 11–12, 1982. This meeting will review the growth and applications of TeX. All TUG members are urged to attend. There will be computer site dependent symposia as well as a general overview of TeX-in-Pascal. We hope also to have a demonstration of TeX.

A preliminary schedule will be mailed to TUG members early in the fall, as soon as a program has been devised. We would like to solicit reports on TeX implementation and usage. Discussion topics which are submitted by September 15 will be considered for inclusion in the preliminary schedule.

Please send such requests to:

Tom Pierce
TeX Users' Meeting
P.O. Box 880
Collins Ferry Road
Morgantown, WV 26505

* * * * * * * * * * *

ASK NOT WHAT TUG CAN DO FOR YOU, ASK WHAT YOU CAN DO FOR TUG!

Patrick Milligan
BNR Inc.

At the recent TEX Implementors' Workshop, there were several discussions (both formal and informal) concerning the future of TEX and the TEX Users Group. The following article reflects my opinions about where we should be headed, and how we can get there.

It seems clear that the widespread acceptance and use of TEX is tied very closely to the success and growth of TUG. Without an effective forum for the interchange of ideas and information, TEX will probably not fulfill its potential as a standard language for computer typography. The TEX Users Group, through TUGboat, has begun to provide such a forum, but in order to function effectively, your assistance is required!

At the time of the Workshop in May, there were over 300 members of TUG. It is not known how many of this number are actual TEX users (as opposed to *potential* users awaiting a working implementation of TEX on their local computer facilities). In addition, it is not known how many TEX users have not yet become paying members of TUG. By definition, the TEX Users Group must have users of TEX in order to be a viable organization. Therefore, the primary goal of TUG should be to encourage and assist the growth of the TEX *user* community. There are several ways that you, as a member of TUG, can help:

1. If you are lucky enough to have a working TEX installation, encourage your local users to join TUG. In addition, share your experiences with the use and/or installation of TEX by sending letters, articles, bugs, and macros to TUGboat.
2. If you have received a version of TEX and are in the process of installing it on your local computer, let TUGboat know about your progress (or lack of progress). News of (temporary) failure is just as important as news of success!
3. If you are waiting for a version of TEX to be available on your flavor of computer architecture, contact your site coordinator to indicate your interest. In this way, you might be able to receive advance notice of a working TEX. Also, you might begin to acquire the necessary hardware for your output devices and begin to build some of the support software necessary to drive such devices.
4. If no one is implementing TEX on your flavor of computer architecture, obtain a copy of TEX-in-Pascal and begin your own installation effort. If you are not a systems programmer, you should be able to interest someone on your local computer staff to assist.

The intent of such communications to TUGboat is to minimize the "reinventing of the wheel". Each potential TEX installer should be able to draw upon a wealth of knowledge on the trials and tribulations of TEX installation. Each novice macro writer should have numerous examples available to learn from. It is frustrating to hear second-hand rumors at TUG meetings or workshops like: "So and So at SRI has a working VAX/UNIX TEX" or "Someone at DEC has a Diablo device interface" or "Somebody at MIT has some nice thesis macros." Just as Don Knuth has shared TEX with the world, it is imperative that you share your TEX experiences with TUG.

Many of TUG's current problems are due to a lack of "critical mass". The porting of Pascal TEX to many architectures, and the availability of output devices and their interfaces has not happened as quickly as anticipated. At the TUG Steering Committee meeting in May, the issues of institutional memberships and TEX support were discussed, but not resolved. The primary obstacles to the institutional memberships were (a) the fear that such fees would inhibit the installation of TEX by small organizations or universities, and (b) the current organization of TUG does not easily allow additional services beyond TUGboat as an enticement to make such fees worthwhile. The bottom line seems to be that there aren't enough TEX installations willing or able to bear the burden of additional services such as TEX support or enhancement. As the number of TEX-in-Pascal installations grows, the direction and functions of TUG will grow also.

Once the first hurdle of providing TEX to a wide base of users is met, there are other challenges for TUG to face. In the area of output device support, there is a strong need for portable device drivers and TEX support tools written in Pascal or some other widely used programming language. Admittedly, standard Pascal does not provide the full set of facilities required to write such device drivers, but most Pascals provide some means of escape or extension to allow full use of the underlying operating system. It would be a useful exercise in portability if large portions of device driver code were written in standard Pascal, with architecture or operating system dependences collected together in one or more system dependent modules (like the SYSDEP code of TEX-in-Pascal). One example of such a program is the Pascal version of DVITYP, written by David Fuchs at Stanford. Pascal TEX itself is an interesting experiment in portability. These examples are just the beginning; much more work needs to be done in this area.

Another direction for TUG growth is in the area of macro packages. Most TeX installations quickly discover that one or more layers of macros are required to insulate their users from "naked" TeX. Many useful macro packages have been presented in TUGboat. Michael Spivak's comprehensive $\mathcal{A}\mathcal{M}\mathcal{S}$-TeX macros have been thoroughly documented in *The Joy of TeX*. However, many more useful and interesting macros have been developed but not contributed to TUG. Also, the issue of portability is applicable to macro packages as well: the use of extended ASCII character sets, font codes, counters and boxes all make the job of merging several macro packages together difficult. Output device dependences may find their way into macros, thus defeating TeX's "device independent" output. It is hoped that Lynne Price, the TUG macro coordinator, may be able to bring some order out of chaos in this area (with your help). Awareness of the portability and modularity issues will assist TeX macro writers; standards and conventions encouraged by TUG will also help.

Closely related to the issues of portable TeX support tools and macro packages is the area of machine readable distribution. A proposed standard for machine independent tape interchange is discussed elsewhere in this issue of TUGboat (page 10). Stanford has attempted to solve this problem for the distribution of TeX-in-Pascal, macros, and fonts. The current organization of site coordinators has solved the problem of distribution between sites using similar computers, through the use of common, operating system dependent tape formats. However, the problem of general, machine independent tape interchange between TeX users who use different computers has not been completely solved. It is important that standards for tape interchange be established, and portable tools developed to support these standards.

One potential area which TUG should explore is the sale of machine readable macros and programs submitted to TUGboat. Having one distribution center for these contributions would be preferable to contacting the author(s) of a particular program or macro package. Receiving one tape from TUG would be easier than requesting tapes from multiple sources, and would be much easier than typing in part or all of a long macro package or program. In addition, TUG would have another source of revenue! This sort of scheme has worked well for the DECUS Library (a part of the DEC Users Society), and for Addison-Wesley's distribution of Ratfor source for the programs in Kernighan and Plauger's *Software Tools*.

In conclusion, it is clear that what you get out of TUG depends on what you are willing to put into it! Without member contributions, there would be no TUGboat. Without volunteers, there would be no TUG Steering Committee. The future for TeX and TUG looks bright, provided we can ease our growing pains (with your help). Before I step down from my soap box, I would like to thank all of you who have made the TeX Users Group and TUGboat possible through your involvement. The staff of the American Mathematical Society deserve special thanks for their hard work and patience.

* * * * * * * * * * *

A PROPOSAL FOR A MACHINE INDEPENDENT TAPE INTERCHANGE STANDARD

Patrick Milligan
BNR Inc.

At the TeX Implementors' Workshop in May, a committee was formed to propose a tape format suitable for machine independent and operating system independent interchange of TeX source files. The members of this committee are:

Nelson Beebe	University of Utah
Patrick Milligan	BNR Inc.
Robert Morris	UMASS/Boston
Susan Plass	Stanford CIT

The motivation behind this proposal is to provide a means of submitting machine readable TeX source to TUGboat (and someday to AMS journals), as well as a means of distributing and exchanging TeX macros and manuscripts. To some extent, the problem of tape interchange formats has been addressed by TUGboat in its ASCII "card image" format (80 characters/record × 100 records/block). The primary problems with such a format stem from TeX's use of the full ASCII character set. The following potential problems exist:

- Not all computer systems support the ASCII character set, and those that do may limit or prohibit the use of ASCII control characters. There are "standard" translations between ASCII and EBCDIC graphic characters, but no such translations exist for control characters. TeX can usually avoid the use of control characters, but as we have seen in recent TUGboat macro packages and in the TeX manual itself, it is tempting to use the "extended" ASCII character sets in use at Stanford, MIT, and CMU if they are available. In addition, TeX's control sequences for negative conditional thin space (\≤) and conditional thin space (\≥) *must* be entered using control characters!

- TeX makes some assumptions about the underlying structure of text files. In particular, it is assumed that a file is organized as a long string of characters which is divided into lines by end-of-line characters, and into pages by form-feeds. On some systems, the structure of text files is either fixed length "card image" records, padded with blanks (and possibly with sequence numbers in columns 73–80), or variable length records rounded to computer word boundaries and padded with blanks or some other filler. In most cases, it is not important to know where the placement of the end-of-line is, or whether the trailing blanks on a line are "real" or supplied by the system. However, if the meaning of blanks or end-of-line characters is changed through the use of the \chcode control sequence, their placement and existence becomes critical. Many powerful techniques presented at the TeXarcana mini-course depend on the ability to redefine space or carriage-return to invoke a control sequence. Arthur Keller's \nofill macro (presented in TUGboat Vol. 2, No. 1) also uses this feature of TeX.
- Another attribute of some text file representations is limited line length. The worst case seems to be the fixed width card image format with sequence numbers. Since TeX allows lines up to 150 characters, unless care is taken, TeX source may overflow the 72 character limit imposed by some systems. Even if a conscious effort is made to limit line length, there are times when it is difficult if not impossible to break a line for fear of introducing a significant space. For example, the \qspace macro in AMS-TeX has one line which is 98 characters long, and it can't easily be broken since the space character has been redefined to be category 12 via \chcode.

Many of the problems listed above must be resolved in the system dependent module of Pascal TeX for each architecture. By definition, our tape interchange format must be independent of the design decisions that were made for a specific implementation of TeX. The best we can do is provide a format that can be transformed into suitable input for Pascal TeX on a given system. It is also hoped that such a transformation is reversible. An additional constraint placed on our tape format is that it should be able to accommodate TeX source containing control characters, significant trailing spaces and carriage returns, and long lines. It is not our place to pass judgment on the use of TeX's somewhat esoteric tricks: We must accept the reality that such features will be used.

In order to meet our constraints of machine independence and compatibility with TeX's idealized notions of text files, we are proposing a tape format which represents a TeX source file as a stream of ASCII characters separated into lines by carriage-return linefeed pairs. This stream of characters will be broken into tape records N bytes long, where N will be chosen such that (1) a tape record will exactly fill an integral number of words on all targeted architectures and (2) N will be large enough to effectively utilize the tape. Suggestions for a good value of N would be greatly appreciated! The last block of the tape should be padded with NULs. In order to avoid problems with "helpful" systems that like to throw away "unwanted" characters, each ASCII character will be represented as two hexadecimal digits.

In order to make this format work, each TeX installer for a given architecture will have to write two programs: One to read such a tape and transform the data into a machine-dependent text file format that TeX will digest, and another program to perform the reverse transformation and output a hex-encoded tape. The design decisions that went into the implementation of the system-dependent module for Pascal TeX will be applicable to these tape utilities.

It is assumed that 9-track tapes will be used, although the hex encoding would work equally well for 7-track tapes (using a 6-bit ASCII subset for each digit). The same coding scheme can be used to transfer files over phone lines if N is chosen to be a reasonable terminal line length.

An added benefit to this format is that it can be used to transfer binary data such as DVI, TFM, and font files with few modifications. In this case, the two hex digits would represent an 8-bit data byte instead of a 7-bit ASCII character.

It seems clear that we need a tape standard that addresses the problems of machine independent information exchange, while still providing the functionality that TeX requires. There are two questions to be asked:

1. Is this the format that we need?
2. Is it worth the effort involved?

Your input is needed to answer these questions. Feedback from those of you who have been actively working on porting Pascal TEX to new architectures is especially welcome. Please respond!

* * * * * * * * * * *

Software

* * * * * * * * * * *

THE FORMAT OF TEX'S DVI FILES VERSION 1

David Fuchs
TEX Project, Stanford University

April 18, 1981

When TEX compiles a document, it produces an output file that contains specifications of how TEX has decided the formatted text should appear in hard copy. These output files are known as '.DVI' files, which stands for 'device independent'. For instance, running TEX and telling it to \input dviinf will cause TEX to look for a file called DVIINF.TEX, read it, and produce an output file called DVIINF.DVI, which is a .DVI file. This document describes the format of .DVI files in detail, giving all the specifications along with examples.

A .DVI file contains information about where characters go on pages. The format is such that there are those who say that almost any reasonable device can be driven by a program that takes .DVI files as input. In particular, a .DVI file can be printed on the Xerox Dover, Xerox Graphics Printer (XGP), Varian, Versatec, Canon and Alphatype at the Stanford CS Dept., depending on what spooler it is passed to.

The .DVI file is a stream of 8-bit bytes, packed in computer words high-order byte first. If the computer word length is not evenly divisible by 8, then the extra bits at the low-order end of each word will be unused. The first byte in a .DVI file is byte number zero, the next is number one, etc. For example, on Stanford's 36-bit word machines, byte number 0 is in the highest order eight bits of the first word in a .DVI file, while byte number 7 is in the twelfth through fifth least significant bits of the second word in the file; and the least significant four bits in every word are zero.

A .DVI file is actually a series of **commands**. A **command** consists of one byte containing the **command**'s unique number, followed by a number (possibly zero) of **parameters** to the **command**. A given **command** always has the same number of **parameters**. These **parameters** may take from one to four bytes each, but a given **parameter** of a given **command** always takes the same number of bytes. Some **parameters** may sometimes be negative, in which case two's complement representation is used. The complete list of **commands**, with a description of all the .DVI **commands** and their **parameters**, is below. The reader is encouraged to refer to the **command** list while reading the various examples in this document.

In the **command** descriptions, a lower case letter with a [bracketed] number following it means that the **command** has a **parameter** that is that number of bytes long. An X3 **command**, for instance, is 3 bytes long, the first byte of which has the decimal value 144, the second and third of which give the distance to move to the right. If the second byte = S and the third = T, then the distance to move is 2^8S+T (but if the high order bit of S is a one, then the distance to move is 2^8S+T-2^{16}, considering S and T as being in the range [0..255]).

The .DVI file contains a number of **pages** followed by a **postamble**. A **page** consists of a BOP **command**, followed by lots of other **commands** that tell where the characters on the page go, followed by an EOP **command**. Each EOP **command** is immediately followed by another BOP **command**, or by the PST **command**, which means that there are no more **pages** in the file, and the remaining bytes in the .DVI file are the **postamble**. Remember that TEX really doesn't have an official knowledge of **page** numbers (although it does print the value of \count0 on your terminal as it outputs each **page** on the assumption that some meaningful number is there), so the only thing that can be said about the ordering of pages in a .DVI file is: The order in which **pages** come in a .DVI file is the same order in which TEX constructed them, which is the same order in which the TEX user specified them. Any blank or nonexistent **page** from a TEX job might not be in the .DVI file at all. If we consider the **page** number to be the value of \count0, then the **page** following **page** number 34 in a .DVI file might well be **page** number —5.

Some **parameters** of .DVI **commands** are **pointers**. A **pointer** is simply a byte number as discussed above. A **pointer** itself is 4 bytes long. For example, a BOP **command**'s last **parameter** (p[4]) is the BOP's **previous page pointer**. This parameter is the number of the byte in which the previous **page**'s BOP command begins. In particular, the *second* **page**'s BOP **command**'s **previous page pointer parameter** (p[4]) is always zero, since the first **page**'s BOP is always in byte zero in a .DVI file. If the first **page** in a .DVI file had only a BOP and EOP **command**, then the *third* **page**'s BOP's **previous page pointer**