Computers can make it easier, quicker, and cheaper to print formal publications; individuals can do what was once practical only for organizations. ... As the computer increases the freedom of writers, so does it increase the responsibility of readers.

John Shore
The Sachertorte Algorithm and other antidotes to computer anxiety,
Penguin Books, 1986, pp. 18, 20

TUGBOAT

THE TEX USERS GROUP NEWSLETTER
EDITOR BARBARA BEETON

VOLUME 7, NUMBER 2 • JUNE, 1986
PROVIDENCE • RHODE ISLAND • U.S.A.

TUGboat

The communications of the TEX Users Group are published irregularly at Providence, Rhode Island, and are distributed as a benefit of membership both to individual and institutional members.

Submissions to TUGboat are for the most part reproduced with minimal editing, and any questions regarding content or accuracy should be directed to the authors, with an information copy to the Editor.

Submitting Items for Publication

The deadline for submitting items for Vol. 7, No. 3, is August 25, 1986; the issue will be mailed in late October.

Manuscripts should be submitted to a member of the TUGboat Editorial Committee. Articles of general interest, those not covered by any of the editorial departments listed, and all items submitted on magnetic tape or as camera-ready copy should be addressed to the Editor, Barbara Beeton.

Contributions in camera copy form are encouraged, as is electronic submission of items on magnetic tape, via electronic mail, or transferred directly to the AMS computer; for instructions, write or call Barbara Beeton.

TUGboat Advertising and Mailing Lists

For information about advertising rates or the purchase of TUG mailing lists, write or call Ray Goucher.

Other TUG Publications

TUG is interested in considering for publication manuals or other documentation that might be useful to the TEX community in general. If you have any such items or know of any that you would like considered for publication, contact Ray Goucher at the TUG office.

TUGboat Editorial Committee

See page 91 for addresses.

Addresses

James Alexander
University of Maryland
Electrical Engineering Dept
College Park, MD 20742
Arpanet: alex@eneevax.umd.edu

Lawrence A. Beck
Grumman Data Systems
R & D, MS D12-237
Woodbury, NY 11797
516-682-8478

Helmut Becker
Rittershausstr. 4
D-5300 Bonn, West Germany
+49 228 211850
UNI15C@DBNRHRZ1.Bitnet

Barbara Beeton
American Mathematical Society
P. O. Box 6248
Providence, RI 02940
401-272-9500
Arpanet: bb@SU-AI

Neil Block
P O Box 8829
Fountain Valley, CA 92728-8829
213-513-4545

Lance Carnes
163 Linden Lane
Mill Valley, CA 94941
415-388-8853

S. Bart Childs
Dept of Computer Science
Texas A & M University
College Station, TX 77843-3112
409-845-5470

Maria Code
Data Processing Services
1371 Sydney Dr
Sunnyvale, CA 94087

John M. Crawford
Computing Services Center
College of Administrative Science
Ohio State University
Columbus, OH 43210
614-422-1741
CSNet: Crawford-J@Ohio-State
BITNet: TSO135@OHSTVMA

Chuck Dupree
Omnicad Corp
1000 Pittsford-Victor Rd
Pittsford, NY 14534
716-385-8500

Maureen Eppstein
Administrative Publication Manager
Stanford University
Encina Hall, Room 200
Stanford, CA 94305
415-497-9254
Arpanet: MVEppstein@SU-Score

Michael J. Ferguson
INRS - Télécommunications
Université du Québec
3 Place du Commerce
Verdun (H3E 1H6), Québec
Canada
514-765-7834

Sylvester Fernandez
Sperry Corporation
Defense Products Group
Sperry Park, P. O. Box 64525
St. Paul, Minnesota 55164-0525
612-456-2222

Jim Fox
Academic Computing Center
HG-45
University of Washington
3737 Brooklyn Ave NE
Seattle, WA 98105
206-543-4320
Bitnet: fox7632@uwacdc

David Fuchs
Department of Computer Science
Stanford University
Stanford, CA 94305
415-497-1646
Arpanet: DRF@SU-Score

Richard Furuta
Department of Computer Science
Univ of Maryland
College Park, MD 20742
301-454-1461
Arpanet: Furuta@Maryland

Peter Gordon
Addison-Wesley Publishing Co
Applications Software
Reading, MA 01867
617-944-3700 x2363

Raymond E. Goucher
TEX Users Group
P. O. Box 9506
Providence, RI 02940
401-272-9500 x232

William Gropp
Dept of Computer Science
Yale University
Box 2158 Yale Station
New Haven, CT 06520
203-436-3761
Arpanet: Gropp@Yale

Alan Hoenig
574 Argyle Rd
Brooklyn, NY 11230
718-856-3696

Kathy Hornbach
Lear Siegler, Inc
Instrument Division
4141 Eastern SE MS 121
Grand Rapids, MI 49508
616-241-8800

Patrick D. Ion
Mathematical Reviews
416 Fourth Street
P. O. Box 8604
Ann Arbor, MI 48107
313-763-6829

Calvin W. Jackson, Jr.
1749 Micheltorena St
Los Angeles, CA 90026
818-356-6245 or 213-660-3257
calvin@cit-20.caltech.edu

Helmut Jürgensen
Dept of Computer Science
Univ of Western Ontario
London N6A 5B7, Ontario,
Canada
519-679-3039
Bitnet: A505@UWOCC1
UUCP: helmut@deepthot

Arthur Keller
University of Texas at Austin
Department of Computer Science
Austin, TX 78712-1188
512-471-7316
ARK@SALLY.UTexas.EDU

Bill Kelly
Academic Computing Center
University of Wisconsin, Madison
1210 W. Dayton Street
Madison, WI 53706
608-262-9501
uwmacc!bllklly@rsch.wisc.edu

John Kennedy
Mathematical Physics
University College Dublin
Dublin 4, Ireland
JKENNEDY@IRLEARN.Bitnet

Karl Kleine
Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14
D-7500 Karlsruhe, West Germany
kleine@germany.csnet
kleine@uka.uucp

Donald E. Knuth
Department of Computer Science
Stanford University
Stanford, CA 94305
Arpanet: DEK@SU-AI

Pierre A. MacKay
University of Washington
Department of Computer Science, FR-35
Seattle, WA 98195
206-543-2386
Arpanet: MacKay@Washington

Rick Mallett
Computing Services
Room 1208 Arts Tower
Carleton University
Ottawa (K1S 5B6), Ontario
Canada
613-231-7145

Richard S. Palais
Department of Mathematics
Brandeis University
Waltham, MA 02154
617-647-2667

Mitch Pfeffer
148 Harbor View South
Lawrence, NY 11559
516-239-4110

Arnold Pizer
Department of Mathematics
University of Rochester
Rochester, NY 14627
716-275-4428

Craig Platt
Depr of Math & Astronomy
Machray Hall
Univ of Manitoba
Winnipeg R3T 2N2
Manitoba, Canada
204-474-9832
platt%uofm-uts.cdn@ubc.csnet

Tom Rokicki
Computer Science Dept
Stanford University
Stanford, CA 94305
Arpanet: Rokicki@SU-Sushi

Barry Smith
Kellerman & Smith
534 SW Third Ave
Portland, OR 97204
503-222-4234

Alan Spragens
SLAC Computing Services
Stanford Linear Accelerator Ctr
Bin 97, P O Box 4349
Stanford, CA 94305
415-854-3300 x2849
Bitnet: Spragens@SLACVM

Ralph Stromquist
MACC
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706
608-262-8821

Rilla Thedford
Intergraph Corporation
One Madison Industrial Park
Huntsville, AL 35807
205-772-6494

Georgia K.M. Tobin
The Metafoundry
OCLC Inc., MC 485
6565 Frantz Road
Dublin, OH 43017
614-764-6087

Joey K. Tuttle
I P Sharp Associates
220 California Avenue
Suite 201
Palo Alto, CA 94306
415-327-1700

Samuel B. Whidden
American Mathematical Society
P. O. Box 6248
Providence, RI 02940
401-272-9500

Hermann Zapf
Seitersweg 35
D-6100 Darmstadt
Fed Rep Germany

General Delivery

Comments on the Format of This Issue

Barbara Beeton

The last issue of TUGboat was noteworthy for its new design, specified by the designer Martha Gannet, implemented by David Kellerman and Barry Smith, and bearing on its cover the image of our worthy vessel drawn by Duane Bibby.

This issue returns (temporarily, it is hoped) to the old "scrapbook" format. There are several reasons, chief among them the desire to get it into members' hands before the summer TUG meeting. A second reason, interacting strongly with the first, is the fact that we at the Math Society have not yet succeeded in obtaining the new fonts, and the time required to install them on the Alphatype typesetter (once the METAFONT files are in hand) is greater than the time available before copy is due to the printer. There should be enough lead time before the next issue goes to press to allow the new style to be accommodated in a leisurely fashion.

In the meantime, please contemplate your reactions to the new design (and the old one too), and let us know what they are. (Cal Jackson has already sent in his comments; see page 98.) At the summer meeting, David Kellerman will give a session describing what was involved in implementing the new design. I hope this will be a catalyst for starting a lively discussion.

Computers & Typesetting Coming Out Party

Barbara Beeton

On May 21, 1986, at the Computer Museum in Boston, Addison-Wesley hosted a reception "in celebration of the completion of TEX, Donald Knuth's Computer Typesetting System", as it said on the poster. The five volumes of *Computers & Typesetting* were on display, fresh from the bindery, and they do look slick. (Attendees who looked a little harder could find an earlier work of Don's in a museum display case — volume 1 of *The Art of Computer Programming*.)

There were other goodies on display as well: MicroTEX, MacTEX (both with previewers and laser printer output), and a 5-day old implementation of METAFONT on an IBM PC/AT. Addison-Wesley intends to be a primary marketer of TEX-related software and documentation, as A-W's president, Donald Hammonds, stated in his introductory remarks. Peter Gordon, Don's editor at A-W, added his comments, and then Don described the history of computers and typesetting, starting with Babbage.

The reception was well attended, with many familiar and unfamiliar faces and names present. Don was accompanied by his wife, Jill, and daughter, Jenny. David Fuchs was there, showing off MicroMETAFONT (MicroTEX is now safely in the custody of A-W's technical group), and David Kellerman and Barry Smith were doing the same with MacTEX. Dave Rodgers was watching over Textset's Preview, and dropping hints about what's to be in the next release. Onlookers included Bart Childs, Sam Whidden, Ray Goucher, Alan Hoenig (whose name tag identified him as TUGboat Associate Editor), Georgia and Rick Tobin, and myself; TUG was well represented.

Peter Gordon and Don Knuth have kindly provided the text of their remarks for publication. I think the following pages will give a bit of the flavor of what was a most festive occasion.

Introducing Donald Knuth and *Computers & Typesetting*

Peter Gordon
Addison-Wesley Publishing Co.
at the Computer Museum
Boston, Massachusetts
May 21, 1986

I am especially delighted to be celebrating here, at the Computer Museum, both the TEX system for computer typesetting and the completed work of its author, Don Knuth. It seems fitting that TEX, which is leading a computer-based revolution in the way books and documents of all sorts are produced, should be presented in a thoughtful, historical context. It seems equally appropriate that Don Knuth should himself be introduced in this setting, given his well-known understanding of

the history of computing and his own important contributions to that history.

In promoting TEX, with the publication of both software and books related to it, and in our use of the TEX system in the production of our own books, we at Addison-Wesley hope to make some contribution to the field as well — a contribution to the future of publishing and communications, and, with that, to the future of science, technology, and education.

Before I go further, I should explain what TEX is, and why people are so excited about it. Simply put, TEX is a computer program. It is a sophisticated piece of software developed by Don at Stanford University, through years of design, testing, and refinement, originally to solve problems he perceived in technical book production, and ultimately to show how the best computer science theory can effectively be translated into practice.

TEX is a tool for typesetting written works with the aid of a computer — articles, reports, proposals, books, you name it — a tool with which writers themselves can specify, through a rich language of commands, exactly what each page will look like when finally printed. Another program developed by Don as a companion to TEX, called METAFONT, even allows users to design the type that will appear on these pages.

The TEX system is particularly useful where the document being written contains mathematical expressions, or where book-like quality in appearance is desirable. These two features — math and beauty, if you will — would alone distinguish TEX from other available computer typesetting systems, and lead to comparison, instead, with more expensive typesetting systems used by professional compositors.

But there is more.

TEX for example, is a portable system, running on a wide range of computers, from micros to mainframes, each implementation fully compatible with the others. This is especially important in mixed computing environments, where a variety of machines is in use — which is almost everywhere in the scientific and technical community.

Related to this portability is TEX's printing device independence. Printed output can be obtained from the same TEX-processed file on everything from a CRT screen or dot matrix printer to a laser printer or even a phototypesetter. A writer, in other words, can proofread his or her text in screen display or local printer output before sending the very same file from which it was produced to a phototypesetter.

I won't go into all the features here. You should really see TEX demonstrated, and ask questions until you learn as much about it as you want.

I do want to point out, though, that TEX is rapidly becoming a standard text processing system in many academic departments and research laboratories throughout the world. It is also gaining increasing recognition for its potential in corporate and industrial, in-house publishing, as well as our own more traditional publishing environments. Addison-Wesley editors, for example, are working with more and more manuscripts prepared with TEX. Among my own authors, Fred Brooks, of *Mythical Man-Month* fame, and Carver Mead, co-author (with Lynn Conway) of the seminal *Introduction to VLSI Systems*, have both converted to TEX.

As early as seven years ago, Gordon Bell perceived the importance of TEX. He wrote then that "Don Knuth's [TEX] is potentially the most significant invention in typesetting in this century. It introduces a standard language for computer typography and in terms of importance could rank near the introduction of the Gutenberg press."

Addison-Wesley first became interested in TEX as an extension of our book publishing relationship with Don dating back many years. As a publisher of many scientific and technical books, however, we soon recognized ourselves the significance of his system for our own business. TEX offered the opportunity to produce such books more quickly and more cheaply than ever before possible, and to provide our authors with increased convenience and facility in developing their works.

We are now deeply involved with TEX in two distinct ways.

First, we are the publishers of TEX. We publish a variety of books related to TEX, including Don's own *Computers & Typesetting* series, about which I'll say more in a minute, and books by other authors. One such book I should mention is *LATEX: A Document Preparation System*, by Leslie Lamport, which describes the "front-end" system he built for TEX, and which he, like Don, placed in the public domain.

We also publish TEX software, as well as other software products related to it. MicroTEX, for example, is a complete implementation of TEX for microcomputers developed by David Fuchs at Stanford. MicroTEX currently runs on the IBM PC family of machines, plus compatibles.

The second way Addison-Wesley is involved with TEX is in the production of our books. Our aim is to learn this new technology as deeply as possible through first-hand experience with it, and thereby

to maintain our position among publishers at the forefront of modern book production technologies.

The publication of Don Knuth's five-volume *Computers & Typesetting* series represents the culmination of his work on TEX and METAFONT. And this, specifically, is what we are celebrating today. You can see the books spread around the room. They were all, of course, typeset, by the author, with TEX.

Volume A is the definitive user's guide and complete reference manual for TEX. This book first appeared in softcover form, and many thousands of copies have already been sold around the world.

Volume B contains the complete source code listings for TEX, and incidentally provides an excellent example of how to write and document a very large program.

Volume C is the user's guide and reference manual for METAFONT, the companion to TEX for font design.

Volume D contains the complete source code listings for METAFONT.

Volume E might become the first coffee-table book in computer typesetting. It graphically depicts over 500 examples of METAFONT programming, the programs that generated all the letters used in the five volumes.

I would like now to introduce Don Knuth. In computer science circles, there would be no need to say anything more. His contribution to the field includes the classic series of books on the *Art of Computer Programming*, about which one reviewer has said, it is as important a work for computer science as Euclid's was for geometry. Don is the recipient of the prestigious Turing Award and National Medal of Science. He is Fletcher Jones Professor of Computer Science at Stanford University.

Don is on sabbatical this year working on a book in theology. If he has been compared to Euclid for his work on the *Art of Computer Programming*, and to Gutenberg for his work on TEX, we can only wonder what the next comparison will be.

With Don today are his wife, Jill Knuth, herself the author of a recently-published book, *Banners without Words*, and daughter Jenny, a student at Brown University. Son John, a student at Stanford, is back home in California minding the computers.

Now, needing no further introduction, here is Don Knuth.

Remarks to Celebrate the Publication of *Computers & Typesetting*

Donald Knuth
at the Computer Museum
Boston, Massachusetts
May 21, 1986

The title of the books we're celebrating today is *Computers & Typesetting*, and since we're meeting here in the Computer Museum I think it's appropriate to point out that computers have been intimately associated with typesetting ever since the very beginning. Anybody who reads about the history of computers will soon learn that many of the key ideas go back to 19th century England, where Charles Babbage designed a so-called Difference Engine and went on from there to plan his Analytical Engine.

Babbage's own machines were never completed, but a Swedish author and publisher named Georg Scheutz read about them and was so fascinated that he and his son Edvard actually built a working difference engine. Thus it was that the first sophisticated computing device came to be built in Sweden. And the most interesting thing, to me at least, was that the output of the Scheutz machine was not punched cards or anything like that; their machine actually produced lead stereotype plates from which books could be printed! Several books were, indeed, printed from the output of this early machine. It was demonstrated in 1856 at the Universal Exposition in Paris, and the souvenir album of that exposition contains the following glowing tribute: "This nearly intelligent machine not only effects in seconds calculations that would demand an hour; it prints the results that it obtains, adding the merit of neat calligraphy to the merit of calculation without possible error."[1] I have copied a page from the first computer-produced book—printed in 1857—so that you can see how far we've come since then. As far as I know, this page is the first extant output of an automatic calculator.[2]

---

[1] Leon Brisse, *Album de l'exposition universelle*, Paris, 1857, p. 194. [Cited in Uta C. Merzbach, *Georg Scheutz and the first printing calculator* (Washington: Smithsonian Institution Press, 1977).]

[2] (Editor's note.) The page, which was displayed among images of METAFONT letterforms, contains columns of figures, neatly aligned and separated by rules, and clearly displays the traces of ink that indicate the edges of pieces of metal type impressed on moistened paper.

I'd also like to say a few words about the history of my own work on computers and typesetting. Last week I went back to my diary of 1977 and found an entry from Thursday, May 5, where it says 'Design of TEX started'. My diary says that I worked intensely on the design all day Thursday, Friday and Saturday; then I went to see *Airport 77* and *Earthquake* to relax! The entry for the following Thursday says: 'Wrote draft report on TEX, stayed up till 5 a.m. typing it into machine'. That weekend I went with my wife on a tour of the Sacramento area with Stanford's Library Associates. We saw many examples of fine printing during that trip, and this encouraged me to read a lot of books about font design during the following week. My diary entry for Saturday, May 21, 1977 — exactly nine years ago today — says that by 5 a.m. that day I had made 'rough drafts of lower case and upper case Roman and italics and digits 0–9'. After a few hours of sleep, I spent the rest of that Saturday writing computer programs to plot curves on a raster. Oh, how little I knew in those days about how difficult it would be to complete this work, which I had sketched out in about two weeks!

Why did I *start* working on TEX in 1977? The whole thing actually began long before, in connection with my books on *The Art of Computer Programming*. I had prepared a second edition of volume 2, but when I received galley proofs they looked awful — because printing technology had changed drastically since the first edition had been published. The books were now done with phototypesetting, instead of hot lead Monotype machines; and (alas!) they were being done with the help of computers instead of by hand. The result was poor spacing, especially in the math, and the fonts of type were terrible by comparison with the originals. I was quite discouraged by this, and didn't know what to do. Addison-Wesley offered to reset everything by the old Monotype method, but I knew that the old way was dying out fast; surely by the time I had finished Volume 4 the same problem would arise again, and I didn't want to write a book that would come out looking like the recent galleys I had seen.

Then a nice thing happened. I was on a committee to revise Stanford's reading list for our department's comprehensive exam, and one of the things we had to do was evaluate a book that Pat Winston had just written about *Artificial Intelligence*. We received galley proofs of that book, and the story we were told was that these galleys had been made on a new machine in Southern California, all based on a discrete high-resolution raster. Apparently one of Winston's students at MIT had flown to Los Angeles with that book on magnetic tape, and the galley proofs we saw were the result. Well, I had had lots of experience with rasterized printing, but only at low resolution, so I thought of it as simply an amusing toy. When I saw these galleys of Winston's book, I was astounded, because the resolution was so good I couldn't tell that the type was actually digital. In fact the digital type looked a lot better than what I had been getting in my own galley proofs.

Digital typesetting means patterns of 0s and 1s, and computer science can be thought of as the study of patterns of 0s and 1s. Therefore, it dawned on me for the first time that I, as a computer scientist, would be able to help solve the printing problem that was worrying me so much. I didn't need to know about metallurgy or optics or chemistry or anything scary like that; all I had to do was construct the right pattern of 0s and 1s and send it to a high-resolution digital typesetter like that machine in Southern California; then I'd have my books the way I wanted them. In other words, the problem of quality printing had been reduced to a problem about 0s and 1s. Therefore it was almost an *obligation* for a computer scientist like myself to study the problem carefully.

Within a week after seeing the galleys of Winston's book, I decided to drop everything else and to work on digital typography. Although Winston unfortunately couldn't be present here today — Pat, I can't thank you enough for having written that book!

Ever since these beginnings in 1977, the TEX research project that I embarked on was driven by two major goals. The first goal was *quality*: we wanted to produce documents that were not just nice, but actually the best. Once upon a time, computers could deal only with numbers; then several years passed when they had numbers and uppercase letters; then they became able to deal with both uppercase and lowercase; then they became capable of working with letters of variable width; and by 1977 there were several systems that could produce very attractive documents. My goal was to take the last step and go all the way, to the finest quality that had ever been achieved in printed documents.

It turned out that it was not hard to achieve this level of quality with respect to the formatting of text, after about two years of work. For example, we did experiments with *TIME* magazine to prove that *TIME* would look much better if it had been done with TEX. But it turned out that the design

of typefaces was much more difficult than I had anticipated; seven years went by before I was able to generate letterforms that I began to like.

The second major design goal was to be *archival*: to create systems that would be independent of changes in printing technology as much as possible. When the next generations of printing devices came along, I wanted to be able to retain the same quality already achieved, instead of having to solve all the problems anew. I wanted to design something that would still be usable in 100 years. In other words, my goal was to arrange things so that, if book specifications are saved now, our descendants should be able to produce an equivalent book in the year 2086. Although I expect that there will be a continual development of "front ends" to TEX and METAFONT, as well as a continual development of "back ends" or device drivers that operate on the output of the systems, I designed TEX and METAFONT themselves so that they will not have to change at all: They should be able to serve as useful *fixed points* in the middle, solid enough to build on and to rely on.

Today I'd like to brag a little, and say that I think that these goals of top quality and technology independence seem to be achieved; and volumes A, B, C, D, E tell everything about how it was done. Today I'm seeing these books for the first time, and I'm happy that all of you can be here to help me celebrate this event. These books are somewhat unusual because they describe themselves: They describe exactly how they were typeset. All of the formatting was done by the TEX system described in volumes A and B. Also every letter and every symbol that appears in all five volumes, as well as on the covers and book jackets, was done by the METAFONT system described in volumes C and D. Volume E tells how I dotted all the i's and crossed all the t's, literally. If copies of these books were sent to Mars, the Martians would be able to use them to recreate the patterns of 0s and 1s that were used in the typesetting. Essentially everything I learned during the past nine years is in here.

All of the methods described in these books are in the public domain; thus anybody can freely use any of the ideas. The only thing I'm retaining control of is the names, TEX and METAFONT: products that go by this name are are obliged to conform to the standard. If any changes are made, I won't complain, as long as the changed systems are not called TEX or METAFONT.

Volumes A and C are user manuals; I tried to write manuals that would suit users at all levels as they grow with the systems. And I also strove for a high standard of excellence in the choice of the quotations from other works that are included at the end of every chapter.

Volumes B and D contain the complete program listings of TEX and METAFONT. These books are specifically for computer scientists, not for casual users, but I'm especially pleased with how they came out because they represent an unexpected payoff of my research. This is something that I had no idea would be possible when I began nine years ago. As I wrote the programs for TEX and METAFONT, I wanted to produce systems that would represent the state of the art in computer programming, and this led to the so-called WEB system of structured documentation. I think that WEB might turn out to be the most important thing about all this research — more important in the long run than TEX and METAFONT themselves — because WEB represents a new way to write software that I think is really better than any other way. Using WEB, it was possible to write programs that are so readable that I think there already are more people who understand the inner workings of TEX than now understand any other system that is as large. Furthermore I think it's fair to claim that WEB has made TEX and METAFONT as portable, as maintainable, and as reliable as any other pieces of software in existence. The programs are now running and producing essentially identical results on almost all large computers; there are thousands of users, yet no bugs have been reported for more than half a year. I think there is at most one more bug in TEX, and I'm willing to pay $20.48 to the first person who finds it. (Next year the reward will double, to $40.96, etc.)[3]

Volumes B and D also contain another innovation that improves on the basic WEB system previously available: Every pair of facing pages has a mini-index on the right-hand page, for quick cross reference to anything that's referred to on either page; this saves a lot of time thumbing through the master index at the end.

In recent years I've been making a pitch for programs as works of *literature*. Although there still is no Pulitzer Prize for the best-written computer programs of the year, I tried to write volumes B

[3] (Editor's note.) A listener asked, how much had it cost to pay off the finders of bugs in the programs and errors in the books? Depending on how many checks were actually cashed, Don estimated the total to be between $2,000 and $5,000. It is doubtful that the checkbook in question is easily balanced.

and D in such a way that I would be a candidate for such a prize if it were actually given! More seriously, I intended these books to be useful to computer scientists for self study as well as for study in college seminars. Volume D, in particular, should make a good text for a group of advanced students.

The fifth volume, volume E, is the most fun of all. I hope you will all open a copy and riffle through the pages, so that you can see what I mean. METAFONT is a computer language that is not very much like any other, so my goal in this book was to provide lots of examples of how METAFONT can be used to produce fonts of reasonably good quality. Over 500 examples appear here; they cover every letter, digit, punctuation mark, and other symbol that was used in printing these books.

The fonts you get from these programs have the general name 'Computer Modern'. My colleague Charles Bigelow has contributed an introduction that talks about Modern fonts in general. The book explains how you can make your own personal variations of the fonts, which are designed with many parameters so that they can be generated in almost limitless variety. At the end of the book there are sample pages that show specimens of 75 standard Computer Modern typefaces; and thousands of additional varieties could be generated with ease.

Even if you don't read the METAFONT programs in this book, I think it's appealing just to look at the pictures of these constructed alphabets,[4] and to 'know' that the program on the page facing each letter was what 'drew' that letter; it's all there. Somehow this gives a satisfying sense of completeness and order.

The most important thing I want to talk about this morning is *HELP*. I had lots of help — literally hundreds of people who volunteered to assist this project in significant ways — beginning with Hans Wolf of Addison-Wesley, who taught me the details of the Monotype systems that had been used to typeset *The Art of Computer Programming* in the 60s. I was especially fortunate in my work on font design to have had extensive help from world leaders like Hermann Zapf and Matthew Carter. Another stroke of luck was to have outstanding research associates like David Fuchs and John Hobby. Furthermore my research project at Stanford had generous financial support, most notably from the National Science Foundation and the System Development Foundation. With so much help, it would have been very hard for my research to fail. And my wife Jill gave the most help of all. (Next month we will celebrate 25 years of marriage!)

One final note: People often ask me why TEX and METAFONT are symbolized in these books by a lion and a lioness. When Duane Bibby first came up with the lion idea, I instinctively felt that it was right, but I never understood exactly why this was, until about a month ago when I was in the Boston Public Library. I passed by the magnificent stone lions on the library's grand staircase, and I thought: "That's it! TEX and METAFONT try to be like these lions, fixtures that support a great library.[5] I love books, and lions represent books!" No wonder I'm so happy when I realize that TEX and METAFONT have already contributed to the making of several dozen books of fine quality; it makes me extremely pleased to think that this research will probably contribute to the making of many more fine books in years to come.

[4] (Editor's note.) The pictures, it was pointed out, were generated separately from the text of the examples, and pasted in. If both the raster images and the text had been incorporated at the same time, it would have exceeded the capacity of the machines used to produce the book.

Comments on Document Design Prompted by the New TUGboat Format

Cal Jackson
California Institute of Technology

I've been looking at the latest issue of TUGboat and wondering if I should comment. I decided that I should. It is unfortunate that I reach such a decision when we have a guest editor; the responsibility has always been there.

I *think* I now have a *little* more understanding about what a designer and a typographer and a compositor are trying to achieve in the sense of basic qualities. Note "little.".

The guest editor idea was (is) fantastic. And, I can't think of better people to do it than Kellerman and Smith. They're serious and demand the best

[5] (Editor's note.) One is also reminded of the lions that grandly guard the entrance to the New York Public Library, which celebrated its 75th anniversary during this same week.

of themselves. Their effort, like Knuth's, is one of people that learn from others rather than invent without regard to prior work. It's a fine piece of work and I hope they can find the time to share the nitty-gritty of the experience with TUG members.

I suggest that all TUG members can learn from this experience. How? Have the issue critiqued by several professional designers, typographers, compositors. Publish their critiques. That modality appears to be the predominant one for training people in the graphic arts. Work is not considered right or wrong, or a consensus sought; it is the exposure to critical review that develops the worker.

I hope that there will be other guest editors. I once suggested to Pierre MacKay that there be a competition among TEX users where a work would be judged by professionals. My objective — provide input to users that would improve their visual literacy of typographic material. I cannot think of a better alternative to that than guest editors and subsequent critique by professionals.

Editor's note: Professional criticism has always been welcome, as have well-thought-out comments and suggestions for improvement in the appearance and utility of TUGboat. There have been presentations at several TUG meetings dealing with design issues, and there is a session scheduled for the Tufts meeting on the creation and implementation of the format used for TUGboat 7, No. 1. If any readers know any designers, typographers or compositors who might be interested in critiquing TEX documents for publication in TUGboat, please forward their names and other relevant information to the Editor.

Software

VAX Language Sensitive Editor Templates and Guide for Use with LATEX

Kathy Hornbach
Lear Siegler/Instrument Division

A *Quick Reference Guide* and VAX/VMS Language Sensitive Editor (LSEDIT) templates have been made available for distribution by TUG. The package includes both the printed Guide and the software, which consists of an LSEDIT language definition for LATEX and several new styles, described below. [The software will be provided on magnetic media; for details, see the current TUG publications list.]

Using LSEDIT and the LATEX language definition, a user, regardless of his/her level of experience, can quickly and easily learn to format complex documents using LATEX. Use of LSEDIT reduces the amount of typing necessary by automatically supplying the user with a set of templates that define the basic structure of a given LATEX style. These templates can be selected and filled in or deleted as appropriate. The novice user will use the templates extensively, while the more experienced user will use the templates as an aid in remembering infrequently used commands or formats.

The default LATEX styles supported by the LSEDIT language definition are: article, report, letter and slides (SLiTEX). Also included are three new styles for LATEX: memo, MIL-STD-490 documents, and book form documents. These new styles are supported by the LSEDIT language definition.

VAX/VMS format HELP library entries are included for most of the features in version 2.09 of LATEX and SLiTEX.

Use of this package requires that LSEDIT be installed.

It would be easy enough for me to natter on about the wonders of the new and improved version of METAFONT: The code for the font you are now reading was only a gleam in my eye in mid-December of 1985; The code for the masthead on this column took less than eight hours for me to write; and so on and so on. I will not belabor the point that METAFONT's way of precisely describing how a character is drawn (rather than simply drawing it) allows the designer literally to go from alpha to omega:

AAΔΔΠΠΩ

At present I wish to focus on the *product*, not the *process*, of METAFONT. Most users of TEX believe (rightly or wrongly) that the *process* of METAFONT is something wholly beyond their requirements; its *product*, on the contrary, is fundamental to the effective use of TEX.

Let me digress briefly to enumerate some of the details that must be included in a good font. The first consideration that comes to mind is the wealth of detail in the images of the characters that make up the font. Curves must be smooth and pleasing to the eye. The ratio of descender to mean-height must 'look right' for the character's representation at a given point size, and must be harmonious from one point size to another; in many of the classic typefaces, this visual harmony is much too complex to be adequately captured by simple scaling from a single template.

dp dp

dp dp

On the left, we have letters from a 48 point Roman font and the 12 point version produced by simply scaling down from it; the ratios of ascender and descender to point size are the same for both sizes. On the right, we have the same 48 point, but a 12 point version that has, in essence, been completely redrawn by METAFONT to preserve the slight but significant change in ascenders and descenders which typically occurs in traditional typefaces to enhance the legibility of text faces. *Completely redrawn*, but done so from the same code; thus, METAFONT minimizes effort while maximizing effectiveness.

Widths and heights of component strokes and of the images themselves show subtle changes of their own. METAFONT, with its potential for attention to such necessary niceties, handles all these imaging considerations with far more finesse than is possible with any other digital design system.

All of these imaging adjustments are well and good; but as the reader knows, digital image information is not used by TEX at all. The details that TEX requires are all contained in separate files, the .tfm files. For example, to set text, we need to have an idea of the amount of space between words; ideally, such space will grow or shrink within acceptable limits depending on the exigencies of the text being set; an error here can make a font the ultimate inconvenience rather than the ultimate in convenience.

It is imperative that the information contained herein be as precise as possible in order to make a good font, a font capable of rising to the highest level of digital typesetting quality, that is, a font suitable for TEX. If the amount of space at the side of each letter's image is ill-advised, typesetting quality deteriorates and legbity is sacrificed. (Notice how dreadful the seven words preceding this parenthetical remark look.)

Moreover, the ability to let these spaces shrink or grow ever so slightly according to the letter's environment, called *kerning*, as in

Toffee not Toffee

or to substitute a slightly different image for a character pair or triple, called *ligatures*, as in

Toffee for Toffee

is a hallmark of the best typesetting. METAFONT has the ability to handle both kerns and ligatures and thus can produce fonts that rise above the level of the merely adequate.

In some cases, such information can be derived 'after the fact' (from existing images) and appended to a digital font; but it is not difficult to imagine that such a shotgun marriage of image information and .tfm information will tend not to result in a happy and harmonious union of the two. Quite simply, the simultaneous creation of image and .tfm information as done with METAFONT produces the best results.

More critically, there are some other subtler bits of information in font metric files that are much more nearly impossible to imagine creating with any tool other than METAFONT. Those are the various tidbits in the math/science and symbols fonts that are crucial to the fine setting of equations and formulæ that is one of TEX's strengths. These fonts must have a wealth of information that controls positioning and even the composition of certain characters (built up curly brackets, square brackets, integrals, radicals).

The modern typographer must now understand that his art has become an interdisciplinary pursuit and involves mathematics and programming skills as well as the traditional design concerns. While most current typographers will fail to adjust to this radically different method of type design, there will be many newcomers who will use METAFONT to contribute the beautiful digital typefaces that TEX needs for unprecedentedly superb typesetting.

Powell, Ohio
24 May 1986

Every character in this column was created using METAFONT version 0.81. Fonts used include a prototype sans serif in book and slant styles, a proto-prototype Century Schoolbook text style, and a chiseled-look headline font.
The original of this document was printed on a Canon LBP-CX with a resolution of 300 dpi.

The TEX Logo in Various Fonts

Donald E. Knuth

According to the plain TEX macro package described in *The TEXbook*,

```
\def\TeX{T\kern-.1667em\lower.5ex
         \hbox{E}\kern-.125emX}
```

is the "official" definition of TEX's logo. But the plain TEX macros are specifically oriented to the Computer Modern fonts. Other typefaces call for variations in the backspacing, in order to preserve the logo's general flavor.

The definition above seems to work satisfactorily with the main seriffed fonts of Computer Modern (i.e., with all sizes of `cmr` and `cmsl` and `cmti` and `cmbx`); but sans-serif types are a different story. Indeed, *The TEXbook* itself gives alternative definitions of '`\TeX`' on pages 418 and 419, one for the font `cmssdc10 at 40pt` used in chapter titles (cf. page 36) and one for the `cmssq` fonts used in quotations at the ends of chapters (cf. page 337).

My purpose in this note is to record the various versions of '`\TeX`' that were actually used to typeset the books in the *Computers & Typesetting* series, so that it will be easy to make forgeries of the particular style used there.

In every case the 'E' has been lowered by `.5ex` (half of the x-height); the only variation is in the amount of backspacing represented by the two `\kern` instructions. Let us therefore consider a "generic" TEX logo to be defined by

```
\def\TeX{T\kern α em\lower.5ex
         \hbox{E}\kern β emX}
```

for some α and β. The following values of (α, β) were actually used in the published volumes:

font family	α	β
`cmr`	−.1667	−.125
`cmsl`	−.1667	−.125
`cmti`	−.1667	−.125
`cmbx`	−.1667	−.125
`cmssdc`	−.2	−.06
`cmssq`	−.2	0
`cmssqi`	−.2	0
`cmss`	−.15	0
`cmssi`	−.2	0
`cmssbx`	−.1	0

(The last three were used only to typeset the jacket copy, not the "real" texts inside. It took a bit of fiddling to get the spacing right.)

I've had little experience with other fonts, but they seem to respond to a similar treatment. For example, my paper on "Literate Programming" in *The Computer Journal* **27** (1984), 97–111, was typeset in a variant of Times Roman, and the standard `\TeX` macro worked fine. The captions and references in that article were set in Univers; for that sans-serif font we used $(\alpha, \beta) = (-.2, 0)$ as in `cmssq`.

Meta-METAFONT: An Exhibit at The Cooper Union, NYC

Alan Hoenig

The Herb Lubalin Study Center of Design and Typography of The Cooper Union (41 Cooper Square, New York, NY 10003) featured an exhibit on METAFONT from March 3 to April 4 of this year. I discovered this fact well after the April 4 termination date, so this is only a second-hand report. The curator of the Lubalin Center, Ms. Ellen Lupton, would like this to be a travelling show, and one reason for this report is to whet the appetites of any institutions who might wish to host this show.

The show concentrated on the humanistic nature of METAFONT, rather than upon its mathematical underpinnings. The exhibit consists of text drawn from a series of articles by Don Knuth, Douglas Hofstadter, and Geoffrey Sampson published in the journal *Visible Language* between winter 1982 and winter 1985, and is illustrated with work done by Neenie Billawala of Stanford and Georgia K.M. Tobin of The Metafoundry.

This exhibit will appear in an abbreviated form from July 12 through July 19 at the Stevens Institute of Technology in Hoboken, NJ, and will also be in Edinburgh, Scotland, later this summer; dates and exact location are unavailable. For further information, contact Ellen Lupton directly at the above address or by phoning her at (212) 254-6300, ext 211. She also solicits institutions who would be interested in guest-hosting the exhibit.

Editor's note: The following quote, and the illustrations on this page, are from the "poster" describing the exhibition; the illustrations were created by Neenie Billawalla, Stanford University, and are typical of her experiments with METAFONT.

> Metafont, as Knuth describes is, has a humanistic mission; its value is not so much in the variety it may offer consumers, but in the discipline it offers designers. The Lubalin Center is not so interested in the final 'look' of Metafont. We are interested in Metafont as an open, unfinished method for making and thinking about letters. The specific language METAFONT is under continuous revision; the concept of a *meta-font* provides a frame for questioning the history and spirit of letters.

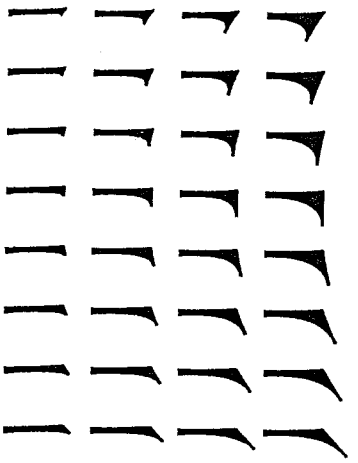
Output Devices

Addenda to the Output Device Charts

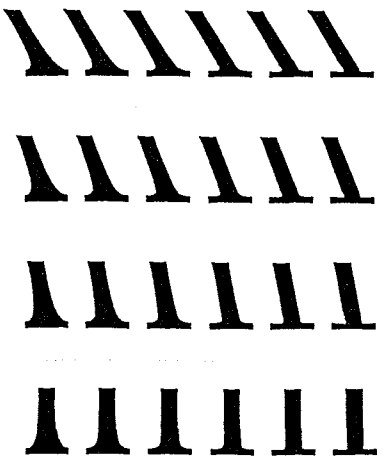
The charts giving information on interfaces between computers and output devices are being omitted from this issue, since almost no changes have been received since their publication in the last issue of TUGboat. The following new information can be added to the existing charts.

MPAE Max-Planck-Institut für Aeronomie
HP 9000 Series 500 with HP LaserJet Plus
Helmut Kopka, [49] 5556-41451
Bitnet: MIO40L@DGOGWDO1

T A&M3 Texas A&M
Commodore Amiga with previewer,
QMS Kiss laser printer
Norman Naugle, 409-845-3104



Terminal serifs with varying heel softness and terminal angle.



Serifs with varying mid-bracket pull and stem slant

Site Reports

TEX on the Amiga

Tomas Rokicki

I am pleased to announce that TEX works on the Commodore Amiga. It first produced a document on March 29, and passed trip shortly thereafter. On a 512K machine, TEX runs with a `memmax` of 22000; on a 1024K or larger machine, a full sized TEX capable of running LATEX or any other macro package runs. A hard disk is certainly nice, but not at all necessary; I've been running TEX on a two drive system with no disk swapping at all. A screen previewer is in the works (it works, the user interface is being improved) and should be available with TEX by the end of June. The screen previewer will be full-featured with dynamically changing magnifications (and zoom capability), access to random pages, and page motion through scroll bars. Various drivers are coming into operation; currently we have a driver for the QMS Kiss laser printer. Because the Amiga is a multitasking machine, you can run your editor, previewer, TEX, and printer driver all concurrently, yielding an impressive TEX environment.

The `virtex` executable comes in at an incredibly small 154,100 bytes; this is smaller than on any other machine I have seen. The final version will be even smaller due to certain optimizations still in progress. It is currently about as fast as a six megahertz PC AT, requiring less than eight seconds per page for `WEB` documents and between four and six seconds per page for normal text. The optimizations we are performing should make it even faster.

TEX, the screen previewer, and various printer drivers are also being ported to the Atari 1040ST. These should be ready shortly. We are also planning a port to the Macintosh. For details on any of these products, contact Norman Naugle at (409) 845-3104 or Tomas Rokicki at (415) 723-1646.

CDC Cyber Site Report

Jim Fox
University of Washington

My last site report indicated that I did not have fonts to distribute. The problem has been corrected. I now have a set of AM style fonts, in `PXL` format, and some utilities to work with them. I expect to investigate the `PK` format and the new CM fonts in the near future.

Prime 50 Series Site report

John Crawford

Things have been going well on the Prime front. They should be even better, as this issue hits the newstands, as my Primos TEX distribution tape will reflect the latest from Stanford. TEX 2.0, the new Computer Modern fonts, and METAFONT 1.0 should all be available.

A technical point of information for Primos users is in order. With Primos revision 19.4.7 and above, my TEX 1.3 should be invoked with the global variable "`.tex_typeahead`" set to `TRUE`. Otherwise, you will likely lose screen output during error displays. I had based my code on the results of a faulty, and later fixed, Primos subroutine. Get the latest port, TEX 2.0, and upgrade out of this workaround.

Typesetting on Personal Computers

The Sperry IT: An IBM AT Compatible

M. Pfeffer and A. Hoenig

The high speed of the Sperry IT makes TEXing and previewing comfortable — far more comfortable than on a PC-class machine. A complete system, using the Sperry system unit (with its fast 40 megabyte hard disk, and 1 Mb of RAM), in conjunction with non-Sperry keyboard, monitor, and Hercules-compatible video card, costs about $3,400.

The System unit: Because TEX is a computation-intensive program, the processing speed of your system is important. Processing speed is determined by two factors: the clock frequency, and the number of wait states required by the system.

The higher the frequency of the system clock, the faster the system. The clock in the original IBM AT ran at 6 MHz; the version released in April 1986 runs at 8 MHz. The Sperry's clock can be switched to 6, 7.16, or 8 MHz.

The second consideration is the number of wait states used: in some machines, the computer's memory can't cough up the information requested by the processor fast enough, forcing the processor to wait. To indicate the delay to the processor, the system introduces one or more wait states. Each wait state degrades the system's performance by 25%.

Both versions of the IBM AT use one wait state. The Sperry uses one wait state in the 6 and 8 MHz settings, but runs with no wait states at the 7.16 MHz speed. This means that the 7.16 MHz speed is the fastest of the Sperry's three speeds, and at this setting, it will out-perform the 8 MHz IBM AT.

For comparison, a 6 MHz AT is two to three times faster than a 4.77 MHz PC; an 8 MHz AT is 33% faster than a 6 MHz AT, while the Sperry, at 7.16 MHz, is 45% faster than a 6 MHz AT.

When using the Textset Preview program, the bottle-neck is disk access. Fortunately, the Sperry is equipped with a fast hard disk (30 ms access time). With this disk, the performance of the preview program becomes acceptable — on slower systems, I find that the performance of the preview program (version 2.1) is uncomfortably slow.

The system unit also includes: two serial ports, one printer port, a 1.2 Mb floppy-disk drive, MS-DOS 3.1, GW-Basic, diagnostic diskette, six available slots (after installing the video card): one 8-bit only slot, and five 8- or 16-bit slots, reset switch, one-year warranty, and manuals (including a setup guide). Technical reference manuals are available.

Important: Sperry currently ships the IT with either a 44 Mb drive made by Miniscribe, or a 40 Mb drive made by CDC. Because I've heard of reliability problems with the Miniscribe 44 Mb drive, specify that the drive in your system not be from Miniscribe. Also specify that the serial number be greater than 414001, ensuring that you receive the latest version of the system: unlike the older version, you can now defeat the serial ports and printer port (necessary if you can't defeat the ports that may be present on other cards you install in the system). The BIOS in the current systems is version 1.48, and corrects problems relating to access of the floppy-disk drives (you can ascertain the BIOS version by running the `romver` program). Also in the current system, the optional math co-processor can run at 8 MHz. (The serial number and drive manufacturer are marked on the outside of the box. A leading digit of '4' in the serial number indicates that the hard drive was installed by Sperry.)

Configuring the System: A limitation in DOS is its inability to access more than 32 Mb on a hard disk. To circumvent this, Sperry allows you to treat the 40 Mb drive as if it were two (or more) drives, each with a separate drive letter. These pseudo-drives will total 40 Mb, and you can specify a size of up to 32 Mb for a pseudo-drive.

The 1 Mb of RAM on the motherboard can be configured in two ways: it can be split into two 512 K segments, with one segment for running programs under MS-DOS, the other segment for use as a ram-disk; or, 640 K can be used for program memory, but then the remaining memory becomes unavailable, and is wasted.

The 512 K/512 K split is handy for users of Personal TEX's Cordata (formally Corona) laser printer driver, as the driver requires a 512 K ram-disk. Squeezing TEX into the other 512 K segment may require some care: you may need to forgo use of RAM-resident software, reduce the number of buffers specified in the `config.sys` file (I normally use 64 buffers on the Sperry), and eliminate device drivers from the `config.sys` file.

Date submitted: May 21, 1986.

Eliminating device drivers has its drawbacks: a device driver is needed by the Sperry to access the part of the hard disk beyond the 32 K DOS limit; also, the `ansi.sys` driver is needed by the Textset Preview program — without it, the screen goes blank when you exit the program, until you type the `cls` command.

To avoid these difficulties, you'll want to increase the system's memory — this requires the installation of a RAM card. AMI manufactures RAM cards that do work properly in the Sperry, and cost about $300 — be sure to request the PAL chip for using the card in a Sperry. These cards do not insert wait states. When ordering RAM to populate the card, specify an access time of 150 ns. (The Mitsuba RAM card, designed for the IBM AT, was rejected by the Sperry's power-on diagnostics. I've also heard that the AST RAM card will not work.)

Compatibility: No problems with TEX-related software or editors. Many games will not run. Jet, a flight simulator, works in Hercules mode. Symphony requires a math co-processor. The current Hayes internal modems will work, but the older versions will not.

Support: The Sperry support center, 800-328-1015, is staffed by knowledgeable, helpful representatives.

My dealings with one service center showed that location's staff to be eager to correct a problem in one unit; unfortunately, they released the unit before resolving the problem.

Sperry's corporate personnel proved to be conscientious and responsive.

Peripherals

The following recommendations on peripherals apply to other AT- and PC-class computers, including the ACS computer I discussed in volume 6, number 3.

Display: The most comfortable display I've used to date is the Panasonic TR-122MYP. It uses a long-persistence, lime-colored (QD) phosphor, which I find more restful than the IBM green display, or any of the amber displays I've used. (The current versions of the Taxan 122 display discussed in vol. 6, no. 3 no longer seems to have the long-persistence I originally admired.)

Display Adapter: Low-cost Hercules-compatible video cards use slow chips, and may have trouble working in fast machines. However, I did test one such card, made by Fortron, which did work in the Sperry. On special request, you can purchase a version of this card with a defeatable printer port — a necessity if you have a card in your system with a printer port that can't be disabled.

(In my system, I have a video card with a custom PAL chip that produces a non-blinking, inverse-video cursor. I find blinking cursors distracting — they make me feel that the "meter's running" as I plod along. I've seen ads for a program that claims to produce a non-blinking cursor.)

Keyboard: The good news is I've found a keyboard with an excellent, solid feel, in the AT layout, with a separate cursor pad as an option. (This keyboard can also be used on an IBM PC or compatibles.) Most keyboards (including those made by Key Tronic, and those sold by Sperry and Compaq) have a mushy feel — as though you're typing on a sponge (this isn't far from the truth: the capacitive keys used in most keyboards have a foam pad beneath each key). IBM's clacky keyboard is the opposite extreme: it feels as if you're breaking through egg shells as you type. One die-hard fan of IBM's keyboard expressed his preference for this new keyboard after a short period of use.

The bad news is that this keyboard isn't available through normal mail-order channels. But you can seek out other brands with the same touch: the keys use a hard-contact switching technology.

While the legends on some cheap keyboards consist of surface markings, which will rub off with use, the keycaps on this keyboard are made by two-shot molding: this means that the plastic that makes up the legend goes through the entire thickness of the top of the keycap. You can spot this type of keycap by gently removing a keycap and looking at its underside; if you can see an inner layer of plastic of the same color as the legend, then the keycap was made by two-shot molding.

Anti-Glare Screen: A new model of the Super-Screen (vol. 6, no. 3) incorporating a grounding mesh is now available. One potential advantage is that by discharging the static build-up, less dust should be drawn to the screen — dust that would gradually degrade the sharpness of the image, unless the mesh is frequently cleaned.

Surge Suppressors: See the results of comparative testing in *PC Magazine*, vol. 5, no. 10, pages 107–146, May 27, 1986.

Diskettes: Elek-Tek, 800-621-1269, sells a box of ten 1.2 Mb Dysan diskettes (a top-rated brand) for $33.

Table Making with INRSTEX

Michael J. Ferguson
INRS-Télécommunications
Montréal

The table commands described here were developed as an integral part of INRSTEX. They have also been implemented as a standalone macro package, `TABLES.TeX`, to run with the standard `Plain` macro package that comes with TEX. This paper was produced using `TABLES.TeX` with the standard TUGboat macros (`TUGBOT.STY`).

The major intent of the table commands is to make the input for tables appear on the editor screen in a reasonable manner. Although this package appears to avoid the basic `\halign` commands and forms of `\cr`, `&`, and `#`, their insertion in tables is not precluded by the structure of the table commands. To use these basic commands will result in some loss of readability but a gain in flexibility.

Below is shown a now classic table (also used in the *The TEXbook*) for demonstrating the table-making skills of a set of macros. The input which produced the table appears in Figure 1. This table is **not** identical to the one in the *The TEXbook*. It is, however, very close.

AT&T Common Stock		
Year	Price	Dividend
1971	41–54	$2.60
2	41–54	2.70
3	46–55	2.87
4	40–53	3.24
5	45–52	3.40
6	51–59	.95*

* (first quarter only)

This example illustrates the basic structure of a table. A `\begintable ... \endtable` pair delineates it, and the row format is defined inside the pair `\begintableformat ... \endtableformat`. The entirely optional *sample row*, immediately after `\endtableformat`, forces all three columns to the same width, by taking the longest word in the table data, `Dividends`, and putting it in a *sample command* `\sa{..}` in each of the three columns. The sample row itself is invisible in the final table. The ~ is a space that is exactly the width of a digit. This permits the data to be lined up in the desired way. The | are separators that insert vertical rules or lines and also allow spaces to be ignored after the data in a column. This makes the columns look

```
\centerline{
\begintable
\begintableformat
\center " \center " \center
\endtableformat
\br{} \sa{Dividend} "\sa{Dividend} "\sa{Dividend}  \er{} %sample line
\-
\br{\:|} \use{3}    AT\&T Common Stock         \er{|}
\-
\br{\:|}     Year |  Price  |   Dividend       \er{|}
\-
\br{\:|}     1971 | 41--54  |    \$2.60        \er{|}
\-
\br{\:|}     ~~~2 | 41--54  |     ~2.70        \er{|}
\-
\br{\:|}     ~~~3 | 46--55  |     ~2.87        \er{|}
\-
\br{\:|}     ~~~4 | 40--53  |     ~3.24        \er{|}
\-
\br{\:|}     ~~~5 | 45--52  |     ~3.40        \er{|}
\-
\br{\:|}     ~~~6 | 51--59  |     ~~.95\rlap*  \er{|}
\-
\br{\:}\use{3} \left{* (first quarter only)}
\endtable }
```

Figure 1. Input for table: AT&T Common Stock

nicer and more readable on the input screen. Other details about this table will be apparent later.

The INRSTEX table commands really use two different types of columns: *rule columns* for insertion (or omission) of vertical rules and *data columns* for the insertion of normal data entries. Generally it is assumed that every pair of data columns is separated by a rule column. In addition, a template for rule columns is provided automatically at the left and right hand sides. The table commands make the insertion or omission of a vertical rule or column as simple as placing a " or a | in the text.

The INRSTEX table commands have been designed to be most useful when the following conventions are obeyed:

- A rule column is specified in the table format using a ". Successive data columns are separated by a rule column. The actual rule is inserted in the text with a | or omitted with a " rather than being put in the format.
- A rule column is appended to each end of the table format statement. In general this means that there is a rule column on both sides of the table. The actual insertion or not of the rule is very easy.
- Struts for maintaining row height are inserted in each row rather than in the table format.

The commands make it very easy to include struts, and insert or omit rules as needed. In fact it is no more difficult than inserting an & in the appropriate place. The advantage of following this discipline is that insertion of partial horizontal lines (rules) is quite easy, as is the exact vertical placement of items in different data columns.

The basic format for building a table is

```
\begintable
<special table forms>
\begintableformat
<table format>
\endtableformat
<first row>
    ...
<last row>
\endtable
```

Each row has the form

```
\br{<..>}  <item> | <item> " <item>
                ...  | <item> \er{<..>}
```

where | means to put in a vertical rule and " means to leave it out of the corresponding rule column. `\br{..}` and `\er{..}` signify the beginning and end of the row respectively; the optional arguments are intended for rules and/or struts if needed. For instance, `\br{\:|}` indicates that this line has a standard strut `\:` and a standard vrule. `\br{\: height 2pt}` indicates that this row has no strut but that the initial vrule has a height of 2pt. The strut could just as easily have been put in the `\er{\:}` instead. The form `\br{}` or `\br{"}` indicates a row with no initial strut or vrule.

The table commands supply a second flavour of vrule that is user definable. This is specified by the command `\|`, which may be used in exactly the same way as |. The *insides* of the `\|` are changed by redefining a command `\sprule`, a *special rule*. The default definition is `\def\sprule{\tvrule {2.5\tr}}` where `\tvrule{<dimen>}` produces a special centered `\vrule` with width `<dimen>`, and `\tr` is TEX default rule thickness, .4pt.

An INRSTEX table format is a template for the table, and corresponds to the normal *preamble* in an `\halign`. In fact any valid `\halign` preamble may be used in the table format, as long as the # are replaced by ##. However, since a rule column is added to the beginning and end of the **format** list for beginning and terminating rules, a repeating field specification, which is indicated by an (extra) & in the format, should **not** start at a vrule indicator or the result will be two adjacent rules.

Simple column formats are indicated by the use of the commands `\left`, `\center`, `\right`, `\math`. and `\displaymath`. The two `\..math` forms are **never** used alone but rather in conjunction with the first three. These forms are **always** separated† by ", which allows for, but does not put in, a vertical rule. The simplest table format, and probably the most useful, is

```
\begintableformat
&\center
\endtableformat
```

This is a repeating format, as indicated by the &. and thus allows for any number of columns. The data in each column is centered.

Another simple table format is

```
\begintableformat
\left " \right " \math\center
\endtableformat
```

This table would have three data columns: The first data column is ordinary text, left justified; the second is right justified text; and the last is centered mathematics. Each pair of data columns is separated by a rule column indicated by ". Rule

† In fact, any field of the form
`&<optional stuff>##<optional stuff>&`
is acceptable.

columns will be added automatically to the left and right hand ends of the specification. The commands `\right`, `\left`, and `\center` insert the appropriate glues and spacing at both sides of the column. The width of this spacing `\tcs`, the *table column separation*, is a `<dimen>`. This may be modified if desired for all tables by putting it in an `\everytable` token list, or for a specific table by placing it in the *special table forms*.

The following is a repeating column specification:

```
\begintableformat
\left " \right " &\math\center
\endtableformat
```

This is identical to the previous example except that the data column format and the implicit rule column added by the table commands, `\math\center "`, is repeated indefinitely. For almost all tables, the only `&` that should appear in the table format is the one indicating the start of a repeating field.

Horizontal rules are specified by `\-` for the standard `\tr` width rule. An `\hrule` the full width of the table should appear all by itself on a line between an `\er` and the following `\br`, or immediately after the table format. Other `\noalign` commands may be inserted as desired. A partial horizontal rule is specified with `\use`, a special version of multispan, and the `\-`.

The table command place some restrictions on the automatic insertion of tabskip glue. Hopefully these will be minor. The initial and final tabskip glues are set to 0pt. This should create no problems. A parameter `\midtabglue` may be modified. This is set just after the first rule column and is turned off at the last rule column. Finally, `\tablespread = {to <dimen>}` will force the table to be `<dimen>` wide, assuming there is enough tabskip glue available.

The instructions in Figure 2 produce the following rather contrived table.

XYZABC						
XYZ			ABC			
X	Y	Z	A	B	C	
372.466	493.7	45	124	489	280	
372.40	493.7	45	124	489	280	
372.	493.7	45	124	489	280	** ii
XY/A		832	abc	774	$\int$	
		qrr	aaa	799		** iii

** i

This example illustrates the power and flexibility of the table commands. Row *i* shows a *modified*

```
\centerline{
\begintable
\def\sprule{\tvrule{5\tr}}
\begintableformat
 &\center
\endtableformat
\-
\br{\:|} \use{6}              XYZABC     \er{|\mst{\:}{3pt}{3pt}\rlap{ \it ** i}}
\-
\br{\:|} \use{3}   XYZ           | \use{3}    ABC    \er{|\mst{\:}{2pt}{2pt}}
\-
\br{\:|}      X   |  Y    |   Z |   A  |   B  |  C  \er{|}
\-
\br{\:|} 372.466  | 493.7 "  45 | 124 \|  489 | 280 \er{|}
\-
\br{\:|} 372.40~  | 493.7 |  45 | 124  |  489 | 280 \er{|}
\br{|}            "       "     | \use{3}\-        \er{|}
\br{\:|} 372.~~~  | 493.7 |  45 | 124  |  489 | 280 \er{|\rlap{ \it ** ii}}
\-
\br{\:|} \use{2}          | 832 | abc  |  774 |$\int$\er{|\mst{$\int$}{0pt}{3pt}}
\br{|} \use{2}\zb{XY/A}   |\use{4}\-                \er{|}
\br{\:|} \use{2}          | qrr | aaa  |  799 |     \er{|\rlap{ \it ** iii}}
\-
\endtable}
```

Figure 2.

strut, \mst, with changed height and depth (the standard strut is 2.5ex high and .9ex deep). \mst is used again in row *iii* where some extra space is added at the bottom of the integral sign, but not at the top. \mst takes three arguments or parameters: The first is the character from which the strut is derived, the second is the additional space on the top of the character and the third is on the bottom. The first two rows of this table had additional space at both the top and bottom while the "integral" row was changed only at the bottom. Fine tuning tables in this way can improve their visual form immensely. Row *iii* demonstrates the use of a \zb or *zero-centered box* for putting items in the center of larger boxes. This row with the \zb has **no** struts. In addition, \zb boxes are centered and have zero height and depth. This means that they take up no vertical space, so that a partial horizontal line or rule may be inserted. Here the \use{4}\- tells the table commands to use 4 (data) columns for the horizontal rule.

Many other variations are possible.

Editor's note: The INRSTEX table commands will be made available through the TEX Users Group in both printed form and on magnetic media. Details can be obtained from the TUG office.

Queries

Editor's note: This is a gentle request to readers who find they can answer any of the queries published in this column. Please send a copy of your answer to the TUGboat editor as well as to the author; all answers will be published in the next issue following their receipt.

WEB System Extensions

I'm working on D. E. Knuth's "WEB System for Structured Documentation". If you have any experiences with or suggestions for improving this software development and documentation tool, please let me know about them. I want to extend the WEB system so that it is possible to use other programming languages than Pascal. If anybody out there uses the WEB system with another text formatting system than TEX or if you have made extensions to the system, I would also like to hear about that. I'm very interested in all your experiences, in what you use WEB for, and in your suggestions for improvement.

Helmut Becker
Rittershausstr. 4
D-5300 Bonn, West Germany
+49 228 211850
UNI15C@DBNRHRZ1.Bitnet

Editor's note: The last issue of TUGboat, **7**#1, contained two articles on this subject: R. M. Damerell, "Error detecting changes to Tangle", pages 22–24, and Wolfgang Appelt and Karin Horn, "Multiple changefiles in WEB", pages 20–21. At least one attempt to build a WEB for C is known to Helmut Jürgensen, who would also be interested in hearing of work in this area, for possible publication in future issues.

This query was originally disseminated via TeXhax and UNIX-TeX, and the following responses have been posted.

WEB (TEX, LATEX etc.) should all conform to ISO-646 (i.e. ASCII) in the following way: In ISO-646 the codes 64, 91–96 and 123–126 are reserved for "national or application oriented use". Many languages use more letters than a–z, these letters are present on keyboards, and the codes above are used for their representation. Every application (e.g. WEB) should be designed so these codes can be used according to the national standards and the documentation should show how this is done.

Staffan Romberger
"Staffan_Romberger_NADA%QZCOM.MAILNET"
@MIT-MULTICS.ARPA

Drama Scripts

Can anyone tell me where I can find macros for formatting a play complete with line numbers and different fonts for characters, text and stage directions?

John Kennedy
Mathematical Physics
University College Dublin
Dublin 4, Ireland
JKENNEDY@IRLEARN.Bitnet

Marking Changes in Revised Documents

I would like to know if any of you have solved the following problem for TEX generated documents:

When revising a document there is a convention where changed lines are identified by a vertical rule in the right margin.

I would like to be able to define macros that can be invoked at the start and end points of a contiguous sequence of changed text that cause the vertical rule to be automatically inserted. The changed text can span paragraphs and pages.

While it is reasonably straightforward to identify the start and end points of the vertical rule, I have not been able to come up with a scheme to accurately determine the height (or depth) of the rule.

If you have looked at this problem and found a solution, I would very much appreciate hearing from you.

Sylvester Fernandez
Sperry Corporation
Defense Products Group
Sperry Park, P. O. Box 64525
St. Paul, Minnesota 55164-0525
(612) 456-2222

Editor's note: This question was posed during the question and answer session at the 1985 TUG meeting, but no solution was presented. One possible approach, requiring action by the output processor, was suggested: At appropriate points in the text invoke `\special` commands which would place beginning and ending points in the right margin; this same information might be passed via `\marks` to the output routine, which could insert additional endpoints at the bottom, and beginpoints at the top, of appropriate pages in the case of spanned text. Although this approach would require a "nonstandard" `.dvi` translator, it avoids the necessity for TEX to know anything about the dimension.

Side-by-Side Source/Output Samples; First-Line Special Handling

I have two questions for TEXperts. (Maybe they've been answered before.)

(1) I wanted to write a little demonstration file , so I wanted a macro "`\showoff` ⟨texstuff⟩ `\endshowoff`" that would perform the texstuff and then give a verbatim listing of texstuff. I couldn't make it work. *The TEXbook*'s macro at the top of p. 382 almost works, but it doesn't give line breaks. Can it be done? One possibility would be to put the texstuff in an auxiliary file and read the file twice. Can it be done without an auxiliary file?

(2) On page *ix* of *Joy of TEX*, Spivak has a paragraph where the first line is in caps-small caps, the remainder in Roman. Can this be done automatically?

James Alexander
University of Maryland
Electrical Engineering Dept
College Park, MD 20742
alex@eneevax.umd.edu

Editor's note: The auxiliary file technique was used to set the tables, input and output, in the article by Michael Ferguson on page 106. The verbatim macros are based on those used by Knuth to set *The TEXbook* (*The TEXbook*, pp. 420–421) and go by the names `\begintt` and `\endtt`. It was found to be expedient to put the `\...tt` commands into the auxiliary file, and to `\let` them `\relax` when the content was to be acted on rather than displayed verbatim; this may not be necessary, but after several ineffectual attempts to input the auxiliary file after entering verbatim mode (yielding at most one blank line and no verbatim text), it seemed most productive to take the easy way out.

This technique might be built into a `\showoff` macro, with one argument giving the name of an auxiliary file, as follows:

```
\def\showoff#1{\begingroup % process the file
  \let\begintt=\relax \let\endtt=\relax
  \input #1 \endgroup
  \bigskip % separate output from input
  \input #1 % print verbatim listing
}
```

The first and last lines (except for comment lines) of the auxiliary file should be `\begintt` and `\endtt` respectively.

In order to preserve line breaks, the verbatim text must be read in while `^M` (the carriage-return character) is active (as it is when `\obeylines`, set on by `\begintt`, is in effect). This may or may not be a suitable convention while actually processing the text, and analysis of this problem should determine the answer to the question "can it be done without an auxiliary file?"

The question about fancy first-line processing has been asked before, in TUGboat 4, no. 2: 80-81, but if any answer has surfaced, we have not seen it.

Customized Editors for TEX

Now that there are TEX and LATEX, what about editors for handling input for these? I would like to compile a list of customized editors. What I expect — well, hope for — is a list of packages for the EMACS family (Gosling, CCA, GNU, Epsilon, ...), but maybe other people have customized their editors also, or even written special editors (VAX/VMS TPU anybody?). Of particular interest are Unix systems and PC's.

When answering, please quote base editor (if any) and version. Also let me know, please, if and how you're willing to make it available. I will compile a list for publication in TUGboat and redistribution via the TEX bulletin boards. If there is sufficient interest, we might even start a TUGboat column.

Karl Kleine
Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14
D-7500 Karlsruhe, West Germany
kleine@germany.csnet
kleine@uka.uucp
...!mcvax!unido!uka!kleine

Editor's note: A set of templates for LATEX, to be used with DEC's Language-Sensitive Editor (LSEDIT), which runs on top of TPU on VMS systems, has been developed at Lear Siegler, Inc. These templates, and a "Quick Reference Guide" for their use, are available through TUG; see the TUG order form and publications list for details.

Post-LATEX Index Formatting

Is there a LATEX addition which formats nice indexes? The amount of manual work remaining after LATEX leaves off is discouraging (after using Digital standard Runoff).

Jim Ludden
Weyerhaeuser Company
Tacoma, Washington 98477
(206) 924-2345

Letters

To the members of TUG:

Now that *The Joy of TEX* has appeared, old friends (and other acquaintances of old) may be wondering why I've stopped using my first name, inserted a new initial, and joined the ranks of those who waggle their academic degrees after them like little pennants, so I'd like briefly to explain.

The idea was for the cover and title page to list the author as "Spivak, M.D., Ph.D.", and thus exude that air of multi-degreed authority so popular in the books fondly being parodied. The clue to the spurious degree would be the Library of Congress Cataloging-in-Publication Data, listing the author as "Spivak, Michael David". (When I say "idea" I really mean something rather more concrete, since all this appeared on the camera copy supplied to the *AMS*.) My mother, to whom the book is dedicated, was looking forward to this bit of academic buffoonery, since my initials were chosen so that the M.D. "would always be in front of your name, instead of after it".

Unfortunately, the *Society*'s doyennes of academic stodginess, impervious to the promptings of humor, clumsily altered this to its present obnoxious form. Other petty changes, sneakily effected behind my back, are too numerous to mention here. (I don't know who wrote the copy for the back of the book, so can't identify the source of the hypocritical remarks about my "lively style" that "makes this an entertaining manual.")

It is customary for the author of a book to accept responsibility for its defects, and endeavor to make corrections and improvements in future editions. In this case, I expressly disavow any such responsibilities. If you find things wrong with the book, please don't tell me, tell the *AMS*; they own it, and apparently feel that they are better qualified than I to decide what should appear in it.

Mike Spivak, *né* Michael David Spivak

Calendar

1986

Tufts University, Medford, Massachusetts

July 14–15 Intermediate TEX
July 16–18 Advanced TEX
July 21–23 TEX Users Group Annual Meeting
July 24–25 TEX Output routines

* * * * *

Illinois Institute of Technology, Chicago, Illinois

July 28–30 Beginning TEX
July 31–Aug 1 Intermediate TEX

* * * * *

Vanderbilt University, Nashville, Tennessee

Aug 4–6 Advanced TEX
Aug 7–8 Macro writing

* * * * *

Stanford University, Stanford, California

Aug 12–14 Beginning TEX
Aug 15–16 Intermediate TEX
Aug 18–20 Advanced TEX
Aug 21–22 Macro writing

* * * * *

College of St. Thomas St. Paul, Minnesota

Aug 18–20 Beginning TEX
Aug 21–22 Intermediate TEX

* * * * *

Aug 18–22 ACM SIGGRAPH 86, Dallas, Texas
Aug 25 TUGboat Volume 7, No. 3: Deadline for submission of manuscripts

University of Illinois, Chicago, Illinois

Sept 15–17 Beginning TEX
Sept 18–19 Intermediate TEX

* * * * *

1987

Jan 30 TUGboat Volume 8, No. 1: Deadline for submission of manuscripts (tentative)

For additional information on items listed above, contact the TUG office (401-272-9500, ext. 232) unless otherwise noted.

TUG Chapters

Members wishing to form TUG Chapters, either in-house or regional, are invited to contact Ray Goucher at the TUG Headquarters for additional information.

Institutional Members

Aarhus Universitet, Det Regionale EDB-Center (RECAU), *Aarhus, Denmark*

Addison-Wesley Publishing Company, *Reading, Massachusetts*

American Mathematical Society, *Providence, Rhode Island*

ASCII Corporation, *Tokyo, Japan*

Baumeister College, *Witten, West Germany*

Bell Northern Research, Inc., *Mountain View, California*

California Institute of Technology, *Pasadena, California*

CALMA, *Sunnyvale, California*

Calvin College, *Grand Rapids, Michigan*

Canon, Inc., Office System Center, *Tokyo, Japan*

Carleton University, *Ottawa, Ontario, Canada*

CDS/WordWorks, *Davenport, Iowa*

Centre Inter-Régional de Calcul Électronique, CNRS, *Orsay, France*

Centro Internacional De Mejoramiento De Maiz Y Trigo (CIMMYT), *México, D.F., Mexico*

City University of New York, *New York, New York*

College of St. Thomas, *St. Paul, Minnesota*

Columbia University, Center for Computing Activities, *New York, New York*

Corbel & Co., *Jacksonville, Florida*

COS Information, *Montreal, P. Q., Canada*

Digital Equipment Corporation, *Nashua, New Hampshire*

Dowell Schlumberger Inc., *Tulsa, Oklahoma*

Edinboro University of Pennsylvania, *Edinboro, Pennsylvania*

Educational Testing Service, *Princeton, New Jersey*

Electricité de France, *Clamart, France*

European Southern Observatory, *Garching bei München, West Germany*

FTL Systems Incorporated, *Toronto, Ontario, Canada*

Geophysical Company of Norway A/S, *Stavanger, Norway*

Grumman Corporation, *Bethpage, New York*

GTE Laboratories, *Waltham, Massachusetts*

Harvard University, *Cambridge, Massachusetts*

Hewlett-Packard Co., *Boise, Idaho*

Humboldt State University, *Arcata, California*

IBM Corporation, Scientific Center, *Palo Alto, California*

Illinois Institute of Technology, *Chicago, Illinois*

Imagen, *Santa Clara, California*

Institute for Advanced Study, *Princeton, New Jersey*

Institute for Defense Analyses, Communications Research Division, *Princeton, New Jersey*

Intergraph Corporation, *Huntsville, Alabama*

Intevep S. A., *Caracas, Venezuela*

Istituto di Cibernetica, Università degli Studi, *Milan, Italy*

Kuwait Institute for Scientific Research, *Safat, Kuwait*

Los Alamos National Laboratory, University of California, *Los Alamos, New Mexico*

Marquette University, *Milwaukee, Wisconsin*

Massachusetts Institute of Technology, Artificial Intelligence Laboratory, *Cambridge, Massachusetts*

Mathematical Reviews, American Mathematical Society, *Ann Arbor, Michigan*

McGill University, *Montreal, Quebec, Canada*

McGraw-Hill, Inc., *Englewood, Colorado*

National Center for Atmospheric Research, *Boulder, Colorado*

National Institutes of Health, *Bethesda, Maryland*

National Research Council Canada, *Ottawa, Ontario, Canada*

Online Computer Library Center, Inc. (OCLC), *Dublin, Ohio*

QMS, Inc, *Mobile, Alabama*

Queens College, *Flushing, New York*

Reed College, *Portland, Oregon*

RE/SPEC, Inc., *Rapid City, South Dakota*

Ruhr Universität Bochum, *Bochum, West Germany*

Sandia National Laboratories, *Albuquerque, New Mexico*

SAS Institute, *Cary, North Carolina*

Schlumberger Offshore Services, *New Orleans, Louisiana*

Schlumberger Well Services, *Houston, Texas*

Science Applications International Corp., *Oak Ridge, Tennessee*

I. P. Sharp Associates, *Palo Alto, California*

Smithsonian Astrophysical Observatory, Computation Facility, *Cambridge, Massachusetts*

Software Research Associates, *Tokyo, Japan*

Sony Corporation, *Atsugi, Japan*

Springer-Verlag, *Heidelberg, West Germany*

Stanford Linear Accelerator Center (SLAC), *Stanford, California*

Stanford University, Computer Science Department, *Stanford, California*

Stanford University, ITS Graphics & Computer Systems, *Stanford, California*

State University of New York, *Stony Brook, New York*

Syracuse University, *Syracuse, New York*

Talaris Systems, Inc., *San Diego, California*

Texas A & M University, Department of Computer Science, *College Station, Texas*

Texas A & M University, Department of Mathematics, *College Station, Texas*

Texas Accelerator Center, *The Woodlands, Texas*

Textset, Inc., *Ann Arbor, Michigan*

TRW, Inc., *Redondo Beach, California*

Tufts University, *Medford, Massachusetts*

TYX Corporation, *Reston, Virginia*

University of British Columbia, *Vancouver, British Columbia, Canada*

University of Calgary, *Calgary, Alberta, Canada*

University of California, Berkeley, Computer Science Division, *Berkeley, California*

University of California, Berkeley, Computing Services, *Berkeley, California*

University of California, San Francisco, *San Francisco, California*

University of Chicago, Computation Center, *Chicago, Illinois*

University of Chicago, Computer Science Department, *Chicago, Illinois*

University of Chicago, Graduate School of Business, *Chicago, Illinois*

University of Delaware, *Newark, Delaware*

University of Glasgow, *Glasgow, Scotland*

University of Groningen, *Groningen, The Netherlands*

University of Maryland, *College Park, Maryland*

University of Massachusetts, *Amherst, Massachusetts*

University of North Carolina, School of Public Health, *Chapel Hill, North Carolina*

University of Oslo, Institute of Informatics, *Blindern, Oslo, Norway*

University of Southern California, Information Sciences Institute, *Marina del Rey, California*

University of Texas at Austin, Physics Department, *Austin, Texas*

University of Texas at Dallas, Center for Space Science, *Dallas, Texas*

University of Washington, Department of Computer Science, *Seattle, Washington*

University of Western Australia, Regional Computing Centre, *Nedlands, Australia*

University of Wisconsin, Academic Computing Center, *Madison, Wisconsin*

University of York, *Heslington, York, England*

Vanderbilt University, *Nashville, Tennessee*

Washington State University, *Pullman, Washington*

Request for Information

The TEX Users Group maintains a database and publishes a membership list containing information about the equipment on which members' organizations plan to or have installed TEX, and about the applications for which TEX would be used. This list is updated periodically and distributed to members with TUGboat, to permit them to identify others with similar interests. Thus, it is important that the information be complete and up-to-date.

Please answer the questions below, in particular those regarding the status of TEX and the hardware on which it runs or is being installed. (Operating system information is particularly important in the case of IBM mainframes and VAX.) This hardware information is used to group members in the listings by computer and output device.

If accurate information has already been provided by another TUG member at your site, you may indicate that member's name, and the information will be repeated.

If your current listing is correct, you need not answer these questions again. Your cooperation is appreciated.

- *Send completed form with remittance* (checks, money orders, UNESCO coupons) to:
  TEX Users Group
  P. O. Box 594
  Providence, Rhode Island 02901, U.S.A.
- *For foreign bank transfers* direct payment to the TEX Users Group, account #002-031375, at:
  Rhode Island Hospital Trust National Bank
  One Hospital Trust Plaza
  Providence, Rhode Island 02903-2449, U.S.A.
- *General correspondence* about TUG should be addressed to:
  TEX Users Group
  P. O. Box 9506
  Providence, Rhode Island 02940-9506, U.S.A.

Name: _____
Home [ ]
Bus. [ ] Address: _____

QTY	ITEM	AMOUNT
	1986 TUGboat Subscription/TUG Membership (Jan.–Dec.) – **North America** New (first-time): [ ] $25.00 each Renewal: [ ] $35.00; [ ] $25.00 – reduced rate if renewed before January 31, 1986	
	1986 TUGboat Subscription/TUG Membership (Jan.–Dec.) – **Outside North America** New (first-time): [ ] $30.00 each Renewal: [ ] $40.00; [ ] $30.00 – reduced rate if renewed before January 31, 1986	
	TUGboat back issues, 1980 (v. 1) 1981 (v. 2) 1982 (v. 3) 1983 (v. 4) 1984 (v. 5) 1985 (v. 6) $15.00 per issue, #1 #1, #2, #3 #1, #2 #1, #2 #1, #2 #1, #2, #3 circle issue(s) desired:	

Air mail postage is included in the rates for all subscriptions and memberships outside North America.
Quantity discounts available on request.

TOTAL ENCLOSED: _____
(*Prepayment in U.S. dollars required*)

* * * *

Membership List Information

Institution (if not part of address):

Title:
Phone:
Network address: [ ] Arpanet [ ] BITnet
[ ] CSnet [ ] uucp

Specific applications or reason for interest in TEX:

My installation can offer the following software or technical support to TUG:

Please list high-level TEX users at your site who would not mind being contacted for information; give name, address, and telephone.

Date:
Status of TEX: [ ] Under consideration
[ ] Being installed
[ ] Up and running since
Approximate number of users:
Version of TEX: [ ] SAIL
Pascal: [ ] TEX82 [ ] TEX80
[ ] Other (describe)

From whom obtained:

Hardware on which TEX is to be used:

Computer(s)	Operating system(s)	Output device(s)

Please answer the following questions regarding output devices used with TEX
if this form has never been filled out for your site, or if you have new information.
Use a separate form for each output device.

Name _____ Institution _____

A. Output device information
   Device name
   Model
   1. Knowledgeable contact at your site
      Name
      Telephone
   2. Device resolution (dots/inch)
   3. Print speed (average feet/minute in graphics mode)
   4. Physical size of device (height, width, depth)

   5. Purchase price
   6. Device type
      [ ] photographic [ ] electrostatic
      [ ] impact [ ] other (describe)

   7. Paper feed [ ] tractor feed
      [ ] friction, continuous form
      [ ] friction, sheet feed [ ] other (describe)

   8. Paper characteristics
      a. Paper type required by device
         [ ] plain [ ] electrostatic
         [ ] photographic [ ] other (describe)

      b. Special forms that can be used [ ] none
         [ ] preprinted one-part [ ] multi-part
         [ ] card stock [ ] other (describe)

      c. Paper dimensions (width, length)
         maximum
         usable
   9. Print mode
      [ ] Character: ( ) Ascii ( ) Other
      [ ] Graphics [ ] Both char/graphics
   10. Reliability of device
      [ ] Good [ ] Fair [ ] Poor
   11. Maintenance required
      [ ] Heavy [ ] Medium [ ] Light
   12. Recommended usage level
      [ ] Heavy [ ] Medium [ ] Light
   13. Manufacturer information
      a. Manufacturer name
         Contact person
         Address

         Telephone
      b. Delivery time
      c. Service [ ] Reliable [ ] Unreliable

B. Computer to which this device is interfaced
   1. Computer name
   2. Model
   3. Type of architecture *
   4. Operating system

C. Output device driver software
   [ ] Obtained from Stanford
   [ ] Written in-house
   [ ] Other (explain)

D. Separate interface hardware (if any) between host computer and output device (e.g. Z80)
   1. Separate interface hardware not needed because:
      [ ] Output device is run off-line
      [ ] O/D contains user-programmable micro
      [ ] Decided to drive O/D direct from host
   2. Name of interface device (if more than one, specify for each)

   3. Manufacturer information
      a. Manufacturer name
         Contact person
         Address

         Telephone
      b. Delivery time
      c. Purchase price
   4. Modifications
      [ ] Specified by Stanford
      [ ] Designed/built in-house
      [ ] Other (explain)

   5. Software for interface device
      [ ] Obtained from Stanford
      [ ] Written in-house
      [ ] Other (explain)

E. Fonts being used
   [ ] Computer Modern
   [ ] Fonts supplied by manufacturer
   [ ] Other (explain)

   1. From whom were fonts obtained?

   2. Are you using Metafont? [ ] Yes [ ] No

F. What are the strong points of your output device?

G. What are its drawbacks and how have you dealt with them?

H. Comments – overview of output device

TEX82 Order Form

The latest official versions of TEX software and documents are available from Maria Code by special arrangement with the Computer Science Department of Stanford University.

Ten different tapes are available. The generic distribution tape contains the source of TEX82, WEB, and the latest (prototype) version of WEB METAFONT, standard test programs for TEX and METAFONT, a few "change" files, the collection of fonts in TFM format, and other miscellaneous materials; a PASCAL compiler will be required to install programs from a generic tape. The TEX distribution tapes include the AMS-TEX, LATEX and HP TEX macro packages; other macro packages will be added as they become available. The special distribution tapes are for the indicated systems only, and should be ordered for these systems instead of a generic tape. Two tapes are PXL font collections covering various magnifications at 200/240 dots/inch and 300 dots/inch respectively.

Each tape will be a separate 1200 foot reel which you may send in advance or purchase (for the tape media) at $10.00 each. Should you send a tape, you will receive back a different tape. Tapes may be ordered in ASCII or EBCDIC characters. All tapes are 1600 bpi.

The tape price of $82.00 for the first tape and $62.00 for each additional tape (ordered at the same time) covers the cost of duplication, order processing, domestic postage and some of the costs at Stanford University. Extra postage is required for first class or export.

Manuals are available at the approximate cost of duplication and mailing. Prices for manuals are subject to change as revisions and additions are made.

Please send a money order or check (drawn on a US bank) along with your order if possible. Your purchase order will be accepted, as long as you are able to make payment within 30 days of shipment. Make checks payable to Maria Code.

The order form contains a place to record the name and address of the person who will actually use the TEX tapes. This should *not* be someone in the purchasing department.

Your order will be filled with the most recent versions of software and manuals available from Stanford at the time your order is received. If you are waiting for a specific release, please indicate this. Orders are normally filled within a few days. There may be periods (like short vacations) when it will take longer. You will be notified of any serious delays. If you want to inquire about your order you may call Maria Code at (408) 735-8006 between 9:30 a.m. and 2:30 p.m. West Coast time.

If you have questions regarding the implementation of TEX or the like, you must take these to Stanford University or some other friendly TEX user.

Now, please complete the order form on the reverse side.

TEX82 Order Form

TAPES:

TEX generic distribution tapes (PASCAL compiler required):

_____ † ASCII format _____ EBCDIC format

TEX distribution tapes in special formats:

_____ † VAX/VMS Backup format _____ IBM VM/CMS format

_____ † DEC 20/Tops-20 Dumper format _____ * IBM MVS format

† Includes WEB METAFONT * Not yet available; call before ordering

Font tapes:

_____ Font library (200/240 dots/inch) _____ Font library for IBM 4250 printer

_____ Font library (300 dots/inch) _____ Font library for IBM 6670 printer

_____ Total number of tapes.

Tape costs: $82.00 for first tape; $62.00 for each additional.

Tape cost = $ _____

Media costs: $10.00 for each tape required. Media cost = $ _____

MANUALS:

_____ TEX: The Program – $28.00 _____ A Torture Test for TEX – $8.00

_____ WEB – $10.00 _____ TEXware – $8.00

_____ TEXbook – $20.00 _____ LATEX – $20.00

_____ BibTEX– $10.00

Manuals cost = $ _____

California orders only: add sales tax = $ _____

Domestic book rate: no charge.
Domestic first class: $2.50 for each tape and each manual.
Export surface mail: $2.50 for each tape and each manual.
Export air mail to Canada & Mexico: $4.00 each.
Export air mail to Europe: $7.00 each.
Export air mail to other areas: $10.00 each.

Postage cost = $ _____

(Make checks payable to Maria Code)

Total order = $ _____

Name and address for shipment:

_____

_____

_____

_____

Person to contact (if different):

_____

_____

_____

_____

Telephone _____

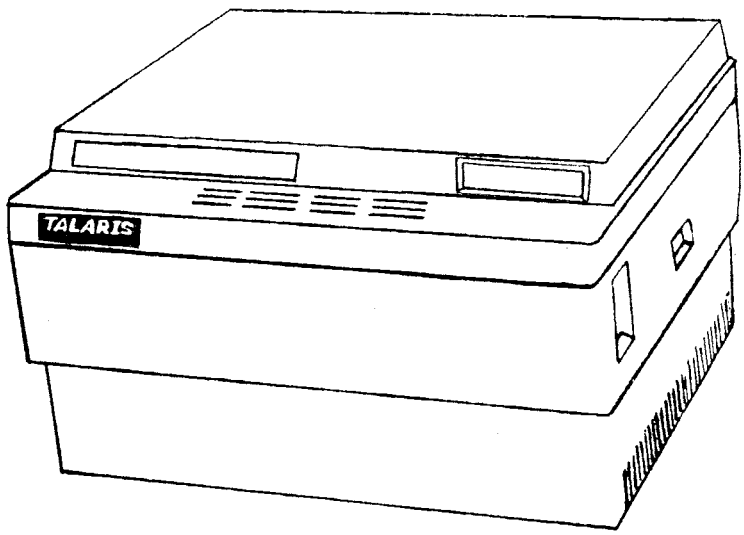
Send to: Maria Code, DP Services, 1371 Sydney Dr., Sunnyvale, CA 94087

TALARIS SYSTEMS INC.®

THE TALARIS 810:

THE MISSING PIECE IN YOUR PERSONAL TYPESETTING SYSTEM

Now that you have TEX™ on your IBM PC or compatible, you need a printer that lives up to TEX's formatting standards—a high-resolution printer with fonts and flexibility. What you don't need is a printer that costs far more than your PC.



Introducing the Talaris 810

The *Talaris 810* is that printer—an 8-page-per-minute, 300 dots-per-inch laser printer that costs just $3,850. Ideal for TEXnicians on a budget.

And Talaris completes your personal typesetting system with fonts and software: TEX PROM font cartridges and a DVI file post-processor.

Talaris's TEX font cartridges

The *Talaris 810* comes with 12 standard word processing fonts. Add one of Talaris's TEX font cartridges and start TEXing. Each font cartridge contains from 16 to 30 TEX fonts. Choose from Plain, Plain + (30 fonts), Dutch Times Roman-like, or Swiss Helvetica-like TEX font cartridges.

Talaris's LoTEX DVI post-processor

LoTEX outputs ANSI 3.64 commands to support absolute character positioning as required by TEX DVI files. LoTEX also provides three options for positioning the page on the paper—centered, uncentered, and TEX standard—as well as options for landscape orientation and printing more than one copy of output.

The Talaris 810 and graphics

Store digitized artwork, forms, and signatures as overlays. Then use TEX's \special command to call the overlays.

Talaris, the personal typesetting software specialists

Talaris has the software to keep your *Talaris 810* printing up to speed. Talaris also offers the *Talaris 810* and LoTEX for VAX/VMS systems. And, for vector graphics and higher volume printing, Talaris offers a family of full-page bit-map laser printers and software.

For more information, call or write Talaris Systems Inc., 5160 Carroll Canyon Road, P.O. Box 261580, San Diego, CA 92126; (619) 587-0787.

This ad was created with TEX and Talaris's Dutch fonts; the original was printed on a Talaris 810 laser printer.

Available Now!

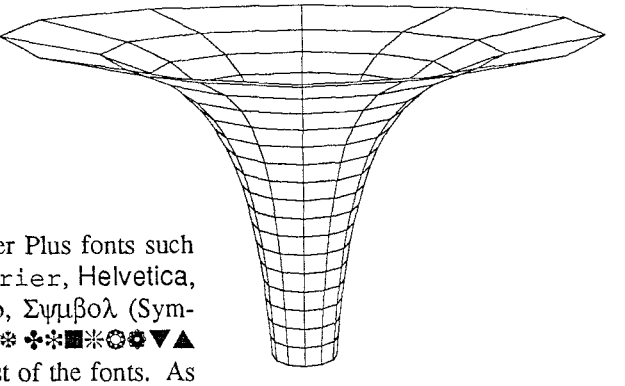
MacTEX™

A complete typesetting system for the Macintosh Plus

MacTEX is the first full version of TEX available for the Macintosh Plus computer. A full implementation, it includes both the PLAIN and LATEX macro packages with LaserWriter and LaserWriter Plus fonts and the standard TEX fonts. With the package, you also get INITEX and all the sources for the macro packages PLAIN and LATEX. With this you can generate your own format files on top of an already existing one, or create a completely new one and select it for loading. And **MacTEX** has passed the "TEX torture test", so you can be sure that the output you get is to the exacting TEX standards.

MacTEX is more than TEX itself. It also contains a built-in multiple file editor and a screen previewer which lets you view the output before printing. Also in the program is the LaserWriter AppleTalk driver to print on a LaserWriter, LaserWriter Plus or any other AppleTalk printer such as a Linotronic 300 (a 2,540 dot per inch laser typesetter). This means that you'll never have to leave the program.

MacTEX will also let you send POSTSCRIPT directly thus allowing effects such as grey text or rotated text. (As you can see above, "Available Now!" is rotated 25 degrees, yet it was set using TEX.) You can also include POSTSCRIPT text files or MacPaint files for graphics. The example on the right was included in printing and the text file containing the POSTSCRIPT code was created using **MacTEX**'s built-in editor.



MacTEX supports all the LaserWriter and LaserWriter Plus fonts such as ITC Avant Garde Gothic, ITC Bookman, Courier, Helvetica, Helvetica-Narrow, New Century Schoolbook, Palatino, Σψμβολ (Symbol), Times-Roman, *Zapf Chancery Medium Italic*, and ✺✪❒❄ ✣❊■❋✪✪▼▲ (Zapf Dingbats) with **bold**, *italic* and ***bold-italic*** for most of the fonts. As well, five LATEX fonts and three math fonts, *amsy, ammi,* and *amex*, are included for full math typesetting and have been converted from pixels into outline form allowing them to be scaled to any size without seeing jagged edges on the letters.

- **Editor:** Standard mouse-driven Macintosh text editor with Cut, Copy, Paste, Clear plus Find, Find Same, Replace, Replace All, and Goto Line.
- **TEX:** Standard TEX with on-screen buttons for error handling.
- **Previewer:** Goto previous page, next page or jump to one at random. View full page (3:1 reduction), full size (1:1) or double size (1:2). Dump the page image to an ImageWriter for proofing.
- **Printer:** Print the *.dvi* file on any LaserWriter on AppleTalk, or download a POSTSCRIPT file. You may also save the *.dvi* file as a POSTSCRIPT text file and download it later.

Also available: **French MacTEX**, containing complete hyphenation tables for the French language.
Font Metric files (*.tfm* files) for other downloadable Adobe Fonts.

Order Now: American Express accepted.

FTL systems Inc.
(416) 487-2142
234 Eglinton Ave. E., Suite 205
Toronto, Ontario
Canada M4P 1K5

This page was set using only **MacTEX** and printed on a LaserWriter Plus.