There's no sense reading directions to something before you understand a little bit about it, because they don't mean anything to you. You have to know enough about something to be confused before directions help.

Andrew A. Rooney,
Pieces of My Mind,
Avon Books, 1985, p. 30

TUGBOAT

THE TEX USERS GROUP NEWSLETTER
EDITOR BARBARA BEETON

VOLUME 6, NUMBER 3 • NOVEMBER, 1985
PROVIDENCE • RHODE ISLAND • U.S.A.

Addresses of Officers, Authors and Others

BECK, Lawrence A.
Grumman Data Systems
R & D, MS B13-237
Woodbury, NY 11797
516-682-8478

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

BIEMESDERFER, Chris
Space Telescope Science Inst
Johns Hopkins University
Homewood Campus
Baltimore, MD 21218
301-338-4864

CAMERON, A. G. W.
Harvard College Observatory
Harvard University
60 Garden St
Cambridge, MA 02138
617-495-5374

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

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

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

CRAWFORD, John M.
Computing Services Center
College of Administrative Science
Ohio State University
Columbus, OH 43210
614-422-1741
CSNet: Crawford-J@Ohio-State
BITNet: TS0135@OHSTVMA

DÉSARMÉNIEN, Jacques
Laboratoire de typographie informatique
7, rue René-Descartes
Université Louis-Pasteur
67084 Strasbourg Cedex, France

DUPREE, Chuck
Datapoint, Inc.
9725 Datapoint Drive, MS N-22
San Antonio, TX 78284
512-699-6342

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

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

FURUTA, Richard
Department of Computer Science
Univ of Maryland
College Park, MD 20742
Arpanet: Furuta@Maryland

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

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

GUENTHER, Dean
Computer Service Center
Washington State University
Computer Sci Bldg, Rm 2144
Pullman, WA 99164
509-335-0411
BITNet: Guenther@WSUVM1

HARRISON, Michael A.
Computer Science Division
571 Evans Hall
University of California
Berkeley, CA 94720
Arpanet: Vortex@Berkeley

HAYASHI, Tsunetoshi
Hokkaido University
Computing Center
Kita 11 Nishi 5, Kitaku, Sapporo
Hokkaido 060, Japan

HENDRICKS, Rick
The Vargas Project
Zimmerman Library
Univ of New Mexico
Albuquerque, NM 87131
505-277-6600

HOENIG, Alan
574 Argyle Rd
Brooklyn, NY 11230
212-856-3696

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

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

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

KELLER, Arthur
University of Texas at Austin
Department of Computer Science
Austin, TX 78712

KELLY, Bill
Academic Computing Center
University of Wisconsin, Madison
1210 W. Dayton Street
Madison, WI 53706
608-262-9501

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

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

LAMPORT, Leslie
Systems Research Center
Digital Equipment Corp
130 Lytton Ave
Palo Alto, CA 94301
415-853-2170
Arpanet: Lamport@decwrl

LEBAN, Bruce
Univ of Massachusetts
P O Box 712
Amherst, MA 01004
413-256-6225
CSnet: Leban@UMass

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

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

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

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

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

ROKICKI, Tom
Department of Computer Science
Stanford University
Stanford, CA 94305
Arpanet: Rokicki@SU-Score

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

SPIVAK, Michael
1660 West Alabama, #7
Houston, TX 77006

SPRAGENS, Alan
SLAC Computing Services
Stanford Linear Accelerator Ctr
Bin 97, P O Box 4349
Stanford, CA 94305
415-854-3300 x2849

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

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

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

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

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

WONNEBERGER, Reinhard
University of Hamburg
Alttestamentliches Seminar
Sedanstr.19
D 2000 Hamburg 13, Fed Rep Germany
B03WBG%DHHDESY3.BITNET@WISCVM.ARPA

ZAPF, Hermann
Seitersweg 35
D-6100 Darmstadt, Fed Rep Germany

TUGboat, the newsletter of the TEX Users Group (TUG), is published irregularly by TUG at Providence, RI, and is distributed as a benefit of membership both to individual and institutional members. Manuscripts should be submitted to a member of the TUGboat Editorial Committee, whose names and addresses are listed inside the front cover. 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, at the general correspondence address. 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.

All general correspondence to TUG should be addressed to TUG, P. O. Box 9506, Providence, RI 02940. All payments to TUG should be addressed to TUG, P. O. Box 594, Providence, RI 02901. Parcel post and items sent by delivery services should be addressed to TUG, c/o American Mathematical Society, 201 Charles St., Providence, RI 02904.

TUG Membership Dues and Privileges

Memberships and Subscriptions

1986 dues for individual members are as follows:

North America:
- New (first-time) members or subscribers: $25.
- Membership and subscription renewals: $35,
  reduced rate of $25 for renewals received before January 31, 1986.

Outside North America (includes air mail postage):
- New (first-time) members or subscribers: $30.
- Membership and subscription renewals: $40,
  reduced rate of $30 for renewals received before January 31, 1986.

Membership privileges include all issues of TUGboat published during the membership (calendar) year. Anyone inquiring about TUG will be sent a complimentary issue of TUGboat, along with a current copy of the membership list and forms for acquiring TEX82, joining TUG and ordering publications available from TUG.

Issues to addresses in the United States are mailed third class bulk, which may take up to six weeks to reach their destinations. If you have not received an issue to which you are entitled, write to TUG at the address given below.

Institutional Membership

1986 Institutional Membership dues for educational organizations are $200; for non-educational, $300. Membership privileges include:
- designating up to 5 persons as individual members,
- special reduced rates for
  participation at TUG meetings and TEX-related courses
  purchase or lease of videotapes.

In addition, institutional members are acknowledged in each issue of TUGboat. For further information, call Ray Goucher at (401) 272-9500, ext. 232.

Submitting Items for Publication in TUGboat

The deadline for submitting items for Vol. 7 (1986), No. 1, will be January 30, 1986; the issue will be mailed in late March. This issue will be guest edited and professionally designed. All contributions may be reformatted, if necessary, to conform to the new design. Electronic submission of items is encouraged, on magnetic tape, via electronic mail, or transferred directly to the AMS computer; for instructions, write or call Barbara Beeton at the address given, (401) 272-9500, ext. 299.

TUGboat Advertising and Mailing Lists

For information about advertising rates or the purchase of TUG mailing lists, write or call the TEX Users Group, Attention: Ray Goucher, P. O. Box 9506, Providence, RI 02940, (401) 272-9500, ext. 232.

General Delivery

Message from the Outgoing President

Pierre MacKay

The August 1985 meeting of the TEX Users Group gave evidence of how fast TEX has matured since its coming out party in 1983. Rather to our surprise there was the announcement that a genuine bug had been found a month or so before and that a new, corrected version of TEX was to be made available as a result. That brings us up to version 1.5 and, since the bug was in a rather obscure corner of the code, it will have affected few users. (It involved the expansion of a `\toks` string in an `\edef` or `\xdef`.) Otherwise, the problems of programming and installation which were once so much a part of every TUG meeting were largely reduced to assurances of success. Even in the case of METAFONT it is no detraction from our deep appreciation of the work currently being done to provide working change files if we note that most of the success with the WEB system, and its constant refinement through experience with TEX, will carry over to METAFONT. We can expect the official release of METAFONT version 1.0 some time in October, and the ports to various systems should come soon after. We look forward most especially to the creation of interactive interfaces to METAFONT which are being developed by our pioneers.

By the time of the TUG meeting, METAFONT had already reached version 0.96 at Stanford, and some external ports were up to version 0.91 or above. The full set of new Computer Modern fonts (these will restore use of the designation CM) was nearing completion. Since Don has a sabbatical planned, he aims to have METAFONT, the CM fonts, and the METAFONT*book* finished when he leaves Stanford. We shall not be left entirely without guidance during his absence, however. For a while at least, David Fuchs will continue to ensure the health of TEX, METAFONT, and their close relatives as he has throughout the entire history of TEX.

The most noticeable change in the focus of interest this year was the evidence of widespread commercial interest in TEX. It was not that there were more vendor exhibits than before, but rather that all the exhibits seemed to cohere into the promise of an ever richer environment for the TEX user. It is still a bit hard to get used to the idea that a program once criticized for its complexity, inaccessibility and sheer size is now being made available on the larger class of personal computer systems, but here it is. Owners of the IBM AT, or the larger versions of the IBM PC have the choice of two different implementations of TEX and an implementation for the Macintosh was also shown. There are more output devices every year, and the quality tends to improve. Access to genuine typesetting is also becoming easier, though there is a great deal to be achieved still in that line. It is clear, however that the TEX community has established itself as a significant market both here and overseas. Each new computer, display system, laser-printer system or typesetter that includes TEX among its capabilities adds to the general range of capabilities for all of us, and we welcome all such developments enthusiastically.

These new developments, however, will bring a clearer responsibility to TUG and its members. As commercial developments proliferate, it will be up to TUG to maintain the standards and quality of the software. One of the most important functions of this organization will be to ensure that anything which is called TEX (or METAFONT) genuinely is TEX (or METAFONT). The mechanism for validation exists, and has been used to establish the credentials of the products mentioned above. I do not at present know of any problems of substandard TEX except for those left over from TEX78, but it would be surprising indeed if none ever developed.

Perhaps even more important than the maintenance of TEX standards will be our insistence on the general character of TEX and METAFONT as public domain software. The TEX system in its largest sense is one of the most significant additions to the library of public domain software that has ever been offered. Every site coordinator is aware of the astonishment that often results from the simple reminder that TEX is and has always been in the public domain. The ports of TEX to small personal systems are examples of the sort of commercial development that TUG must support and encourage, but we must never forget that TEX was given to us free. We are the beneficiaries of seven years of intense development effort, the results of which have been distributed throughout the computing world without restrictions. We can best show our appreciation by continuing to enhance the TEX environment with supporting contributions of free public domain software.

I should like to close this message with a mention of the special support that has been given to TUG by Kellerman and Smith, who will be

offering the Macintosh version of TEX. They have already made other contributions to support TUG, and have now arranged to set aside a royalty on each sale for the further support of TUG. In addition, they have offered to fund the first of the Donald Knuth scholarships for support of attendance at a TEX training course in the coming year. I echo here the deep appreciation that was expressed by the the TUG membership at the business meeting this August.

From the President

Bart Childs

I am proud to accept the challenge of this office and humbly hope that I can contribute to an organization that is dedicated to high ideals. We (the TEX community) are grateful to Don Knuth for his gift of TEX and METAFONT and his placing them in the public domain. Many of us are aware of systems developed under government sponsorship where `revision 0.0001` gets placed in the public domain and `revision 0.0001+` magically appears from some new corporation. As we evolve from what Michael Spivak called "a happy band of anarchists" to our future, I hope that we can keep some of the openess that has been characteristic of our community. I feel a duty to thank Don, David Fuchs, and the whole TEX Project again.

The '85 TUG meeting at Stanford was a great success in spite of the election of someone from TEXas as the new president. Don had obviously laid in wait for the chance for the 'pun'. Still, the class with which he made it was enjoyed by all. I wish to make a few observations on the meeting.

I was surprised by the fact that there was a slight decrease in attendance at the meeting. Many of us have discussed some of the possible causes: the economy, previous meetings have all been at Stanford, ... etc. I will not dwell on that, but, will go on to the things that I think are important for our future.

We will undoubtedly have many pressures from the commercial world that we will affect. I welcome the input of the commercial communities and hope that we can use them as a resource to further our mutual goals. Several people mentioned what they considered "competitive" statements and attitudes by some of the *commercial* representatives. I acknowledge that this is a fair interpretation. I think the comments were made in a professional manner. We must be sure that we don't become a forum whose apparent primary function is a *sales activity*. We shall strive to remain a forum for function, vendor independence, and other high goals while paying appropriate attention to the pragmatism of reality.

Several items reported at the meeting are particularly noteworthy:

1. TEX 1.5 was announced. One error, which could be coded around, was corrected.
2. METAFONT 0.91 was announced (and now 0.96 is out). *Is there any other system which has as many users and as few errors reported as TEX and METAFONT?*
3. TEX is on almost all major systems and is now appearing on many micros (most notably the PC and the Mac).
4. TEX has been mechanically converted to the "C" language at two or three different institutions.
5. TEX is actively used in several different languages.
6. Leslie Lamport delivered the `DVI` files for the LATEX manual while we were meeting. An A-W representative said that the book should be out before you are reading this.
7. TUG will sponsor **Donald E. Knuth** scholarships for end users (for example, secretaries). Kellerman and Smith have guaranteed us the first one.

As your elected leader, I will be active in at least the following areas:

1. Society representation. The AMS has extended me an invitation to present a one hour talk on TEX at the joint meeting with MAA in New Orleans in January 1986. (A hard one to accept!)
2. TEX distribution. Our lab will continue to be the Data General distribution point. There were several conversations about the best format of distributions. I will pursue some efforts to reach accord on the format of a minimum standard and accompanying documentation. Barry Smith and others have discussed this topic and I personally feel these decisions will further the understanding of TEX and its relations.
3. Open communications. I will strive to always prepare this column. We frequently need several views of certain items. Members can take it upon themselves to propose these items and get several members to respond. New

members may wish to communicate the item to the editor or myself for setup of the 'panel'.

Happy TEXing.

Statement of Principles

One of the important decisions taken at the August meeting was that it was time for the TEX Users Group to incorporate. Only by incorporation can we properly confirm our status as a non-profit organization and ensure legal stability and appropriate tax status. As part of the discussion concerning incorporation and tax status, the steering committee considered the necessity of making a general statement which would set down our perception of the purpose of the organizion and our relationship to it. We offer this text of our statement to the general membership of TUG as an informal statement of principles.

1. The primary aim of the TEX Users Group (TUG) is to promote the development and use of public-domain software relating to the TEX system.
2. In addition, TUG encourages commercial development relating to TEX and METAFONT wherever such development contributes to TUG's primary aim.
3. Financial support for TUG's activities derives from membership dues, attendance fees from meetings, fees for courses of instruction, private and corporate grants and donations, and other sources as approved by the Steering Committee of TUG.
4. To avoid any real or apparent conflict of interest, all members of the TUG Steering Committee undertake that they shall make no use of their position on that committee for personal advancement and shall make no private use of information acquired by the Steering Committee unless and until such information has been published to the general membership of TUG.
5. No member of the TUG Finance Committee shall concurrently serve as a direct employee of TUG.

Pierre MacKay

Software

Packed (PK) Font File Format

Pixel files, the output of METAFONT79, are now considered obsolete; their use is being discouraged. This is not surprising; as a font file format, they are rather limiting. They tend to be too large, do not include device character width information, and only allow 128 characters. Yet, the generic font file format offered in their place (with new METAFONT) has some other limitations. GF files tend to be almost as large as the pixel files they replace, and the character width data is separated from the raster data. A new format, called the packed, or PK format, is therefore being introduced.

This new file format is less than half the size of the GF format. Since input/output time usually dominates execution time when reading font files, the smaller size can also lead to a performance improvement. The new format is easier to interpret than the GF format. The minimum bounding box for the bitmaps is supplied; the horizontal and vertical size do not need to be checked against the actual bits of the character. The raster data and width data for each character is given in the same character 'packet'. Finally, the font parameters (checksum, design size, etc.) are given at the beginning of the font file, rather than at the end, so random file access is not needed.

An additional advantage is that the length of each character packet is given at the beginning of the packet. This means that the character packets can be directly stored into the memory of a driver without interpretation, and they can be interpreted on a demand basis. A recent sampling of font usage at Texas A&M University on their Electrical Engineering VAX showed that 18 characters per declared font were used on the average. Thus, a driver using the packed format might not need to interpret 110 of the 128 characters in a font, on the average.

The use of the packed file format does impose an extra step of processing between METAFONT and the driver, however. Also, the packed file format is based to a large extent on the four bit nybble rather than the eight bit byte, so individual bytes often need to be split as the file is being read in. In addition, the generic font format allows **special**s and **numspecial**s within character raster definitions; the packed file format does not. METAFONT never generates **special**s inside of character raster definitions, so this should not pose any problems.

The size improvement of packed files over pixel files is almost five to one for a large collection of fonts. 323 fonts at three hundred dots per inch and various magnifications were reduced by 79% when converted to the packed form. This makes the packed format ideal for microcomputer systems or any system where disk space is at a premium.

The packed file format is part of the METAFONTware, with which it will be distributed. Programs currently on the distribution are GFtoPK, which converts generic font files to packed files; PKtoPX, which converts packed files to pixel files; PXtoPK, which converts pixel files to packed files, and PKtype, which lists and verifies a packed file.

The rest of this article was extracted from PKtype, and is a full description of the packed file format.

1. Packed file format. The packed file format is a compact representation of the data contained in a GF file. The information content is the same, but packed (PK) files are almost always less than half the size of their GF counterparts. They are also easier to convert into a raster representation because they do not have a profusion of *paint*, *skip*, and *new_row* commands to be separately interpreted. In addition, the PK format expressly forbids **special** commands within a character. The minimum bounding box for each character is explicit in the format, and does not need to be scanned for as in the GF format. Finally, the width and escapement values are combined with the raster information into character 'packets', making it simpler in many cases to process a character.

A PK file is organized as a stream of 8-bit bytes. At times, these bytes might be split into 4-bit nybbles or single bits, or combined into multiple byte parameters. When bytes are split into smaller pieces, the 'first' piece is always the most significant of the byte. For instance, the first bit of a byte is the bit with value 128; the first nybble can be found by dividing a byte by 16. Similarly, when bytes are combined into multiple byte parameters, the first byte is the most significant of the parameter. If the parameter is signed, it is represented by two's-complement notation.

The set of possible eight-bit values are separated into two sets, those that introduce a character definition, and those that do not. The values that introduce a character definition comprise the range from 0 to 239; byte values above 239 are interpreted commands. Bytes which introduce character

definitions are called flag bytes, and various fields within the byte indicate various things about how the character definition is encoded. Command bytes have zero or more parameters, and can never appear within a character definition or between parameters of another command, where they would be interpreted as data.

A PK file consists of a preamble, followed by a sequence of one or more character definitions, followed by a postamble. The preamble command must be the first byte in the file, followed immediately by its parameters. Any number of character definitions may follow, and any command but the preamble command and the postamble command may occur between character definitions. The very last command in the file must be the postamble.

2. The packed file format is intended to be easy to read and interpret by device drivers. The small size of the file reduces the input/output overhead each time a font is defined. For those drivers that load and save each font file into memory, the small size also helps reduce the memory requirements. The length of each character packet is specified, allowing the character raster data to be loaded into memory by simply counting bytes, rather than interpreting each command; then, each character can be interpreted on a demand basis. This also makes it possible for a driver to skip a particular character quickly if it knows that the character is unused.

3. First, the command bytes shall be presented; then the format of the Character definitions will be defined. Eight of the possible sixteen commands (values 240 through 255) are currently defined; the others are reserved for future extensions. The commands are listed below. Each command is specified by its symbolic name (e.g., *pk_no_op*), its opcode byte, and any parameters. The parameters are followed by a bracketed number telling how many bytes they occupy, with the number preceded by a plus sign if it is a signed quantity. (Four byte quantities are always signed, however.)

pk_xxx1 240 k[1] $x[k]$. This command is undefined in general; it functions as a $(k+2)$-byte *no_op* unless special PK-reading programs are being used. METAFONT generates *xxx* commands when encountering a **special** string. It is recommended that x be a string having the form of a keyword followed by possible parameters relevant to that keyword.

pk_xxx2 241 k[2] $x[k]$. Like *pk_xxx1*, but $0 \le k < 65536$.

pk_xxx3 242 k[3] $x[k]$. Like *pk_xxx1*, but $0 \le k < 2^{24}$. METAFONT uses this when sending a **special** string whose length exceeds 255.

pk_xxx4 243 k[4] $x[k]$. Like *pk_xxx1*, but k can be ridiculously large; k musn't be negative.

pk_yyy 244 y[4]. This command is undefined in general; it functions as a five-byte *no_op* unless special PK reading programs are being used. METAFONT puts *scaled* numbers into *yyy*'s, as a result of **numspecial** commands; the intent is to provide numeric parameters to *xxx* commands that immediately precede.

pk_post 245. Beginning of the postamble. This command is followed by just enough *pk_no_op* commands to make the file a multiple of four bytes long; zero through three are usual, but four are also allowed. This should make the file easy to read on machines which pack four bytes to a word.

pk_no_op 246. No operation, do nothing. Any number of *pk_no_op*'s may appear between PK commands, but a *pk_no_op* cannot be inserted between a command and its parameters, between two parameters, or inside a character definition.

pk_pre 247 i[1] k[1] $x[k]$ *ds*[4] *cs*[4] *hppp*[4] *vppp*[4]. Preamble command. Here, i is the identification byte of the file, currently equal to 89. The string x is merely a comment, usually indicating the source of the PK file. The parameters *ds* and *cs* are the design size of the file in $1/2^{16}$ points, and the checksum of the file, respectively. The checksum should match the TFM file and the GF files for this font. Parameters *hppp* and *vppp* are the ratios of pixels per point, horizontally and vertically, multiplied by 2^{16}; they can be used to correlate the font with specific device resolutions, magnifications, and 'at sizes'. Usually, the name of the PK file is formed by concatenating the font name (e.g., amr10) with the resolution at which the font is prepared in pixels per inch multiplied by the magnification factor, and the letters PK. For instance, amr10 at 300 dots per inch should be named AMR10.300PK; at one thousand dots per inch and magstephalf, it should be named AMR10.1095PK.

4. We put a few of the above opcodes into definitions for symbolic use by this program.

```
define pk_id = 89
        { the version of PK file described }
define pk_xxx1 = 240  { special commands }
define pk_yyy = 244
        { numspecial commands }
define pk_post = 245  { postamble }
define pk_no_op = 246  { no operation }
define pk_pre = 247  { preamble }
```

5. The PK format has two conflicting goals; to pack character raster and size information as compactly as possible, while retaining ease of translation into raster and other forms. A suitable compromise was found in the use of run-encoding of the raster information. Instead of packing the individual bits of the character, we instead count the number of consecutive 'black' or 'white' pixels in a horizontal raster row, and then encode this number. Run counts are found for each row, from the top of the character to the bottom. This is essentially the way the GF format works. Instead of presenting each row individually, however, let us concatenate all of the horizontal raster rows into one long string of pixels, and encode this row. With knowledge of the width of the bit-map, the original character glyph can be easily reconstructed. In addition, we do not need special commands to mark the end of one row and the beginning of the next.

Next, let us put the burden of finding the minimum bounding box on the part of the font generator, since the characters will usually be used much more often than they are generated. The minimum bounding box is the smallest rectangle which encloses all 'black' pixels of a character. Let us also eliminate the need for a special end of character marker, by supplying exactly as many bits as are required to fill the minimum bounding box, from which the end of the character is implicit.

Let us next consider the distribution of the run counts. Analysis of several dozen pixel files at 300 dots per inch yields a distribution peaking at four, falling off slowly until ten, then a bit more steeply until twenty, and then asymptotically approaching the horizontal. Thus, the great majority of our run counts will fit in a four-bit nybble. The eight-bit byte is attractive for our run-counts, as it is the standard on many systems; however, the wasted four bits in the majority of cases seems a high price to pay. Another possibility is to use a Huffman-type encoding scheme with a variable number of bits for each run-count; this was rejected because of the overhead in fetching and examining individual bits in the file. Thus, the character raster definitions in the PK file format are based on the four-bit nybble.

6. The analysis of the pixel files yielded another interesting statistic: fully 37% of the raster rows were duplicates of the previous row. Thus, the PK format allows the specification of repeat counts, which indicate how many times a horizontal raster row is to be repeated. These repeated rows are taken out of the character glyph before individual rows are concatenated into the long string of pixels.

For elegance, we disallow a run count of zero. The case of a null raster description should be gleaned from the character width and height being equal to zero, and no raster data should be read. No other zero counts are ever necessary. Also, in the absence of repeat counts, the repeat value is set to be zero (only the original row is sent.) If a repeat count is seen, it takes effect on the current row. The current row is defined as the row on which the first pixel of the next run count will lie. The repeat count is set back to zero when the last pixel in the current row is seen, and the row is sent out.

This poses a problem for entirely black and entirely white rows, however. Let us say that the current row ends with four white pixels, and then we have five entirely empty rows, followed by a black pixel at the beginning of the next row, and the character width is ten pixels. We would like to use a repeat count, but there is no legal place to put it. If we put it before the white run count, it will apply to the current row. If we put it after, it applies to the row with the black pixel at the beginning. Thus, entirely white or entirely black repeated rows are always packed as large run counts (in this case, a white run count of 54) rather than repeat counts.

7. Now let us turn our attention to the actual packing of the run counts and repeat counts into nybbles. There are only sixteen possible nybble values. We need to indicate run counts and repeat counts. Since the run counts are much more common, we will devote the majority of the nybble values to them. We therefore indicate a repeat count by a nybble of 14 followed by a packed number, where a packed number will be explained later. Since the repeat count value of one is so common, we indicate a repeat one command by a single nybble of 15. A 14 followed by the packed number 1 is still legal for a repeat one count, however. The run counts are coded directly as packed numbers.

For packed numbers, therefore, we have the nybble values 0 through 13. We need to represent the positive integers up to, say, $2^{31}-1$. We would like the more common smaller numbers to take only one or two nybbles, and the infrequent large numbers to take three or more. We could therefore allocate one nybble value to indicate a large run count taking three or more nybbles. We do this with the value 0.

8. We are left with the values 1 through 13. We can allocate some of these, say *dyn_f*, to be one-nybble run counts. These will work for the run counts 1 .. *dyn_f*. For subsequent run counts, we will use a nybble greater than *dyn_f*, followed by a second nybble, whose value can run from 0 through 15. Thus, the two-byte nybble values will run from $dyn\_f+1\,..\,(13-dyn\_f)*16+dyn\_f$. We have our definition of large run count values now, being all counts greater than $(13-dyn\_f)*16+dyn\_f$.

We can analyze our several dozen pixel files and determine an optimal value of *dyn_f*, and use this value for all of the characters. Unfortunately, values of *dyn_f* that pack small characters well tend to pack the large characters poorly, and values that pack large characters well are not efficient for the smaller characters. Thus, we choose the optimal *dyn_f* on a character basis, picking the value which will pack each individual character in the smallest number of nybbles. Legal values of *dyn_f* run from 0 (with no one-byte run counts) to 13 (with no two-byte run counts).

9. Our only remaining task in the coding of packed numbers is the large run counts. We use a scheme suggested by D. E. Knuth which will simply and elegantly represent arbitrarily large values. The general scheme to represent an integer i is to write its hexadecimal representation, with leading zeros removed. Then we count the number of digits, and prepend one less than that many zeros before the hexadecimal representation. Thus, the values from one to fifteen occupy one nybble; the values sixteen through 255 occupy three, the values 256 through 4095 require five, etc.

For our purposes, however, we have already represented the numbers one through $(13-dyn\_f)*16+dyn\_f$. In addition, the one-nybble values have already been taken by our other commands, which means that only the values from sixteen up are available to us for long run counts. Thus, we simply normalize our long run counts, by subtracting $(13-dyn\_f)*16+dyn\_f+1$ and adding 16, and then representing the result according to the scheme above.

10. The final algorithm for decoding the run counts based on the above scheme might look like this, assuming a procedure called *pk_nyb* is available to get the next nybble from the file, and assuming that the global *repeat_count* indicates whether a row needs to be repeated. Note that this routine is recursive, but since a repeat count can never directly follow another repeat count, it can only be recursive to one level.

```
function pk_packed_num: integer;
var i,j,k: integer;
begin i ← get_nyb;
if i = 0 then
  begin repeat j ← get_nyb; incr(i);
  until j ≠ 0;
  while i > 0 do
    begin j ← j * 16 + get_nyb; decr(i);
    end;
  pk_packed_num ← j-15+(13-dyn_f)*16+dyn_f;
  end
else if i ≤ dyn_f then pk_packed_num ← i
  else if i < 14 then pk_packed_num ←
          (i - dyn_f - 1) * 16 + get_nyb + dyn_f + 1
    else begin if repeat_count ≠ 0 then
        abort('Extra repeat count!');
      if i = 14 then
        repeat_count ← pk_packed_num
      else repeat_count ← 1;
      send_out(true, repeat_count);
      pk_packed_num ← pk_packed_num;
      end;
end;
```

11. For low resolution fonts, or characters with 'gray' areas, run encoding can often make the character many times larger. Therefore, for those characters that cannot be encoded efficiently with run counts, the PK format allows bit-mapping of the characters. This is indicated by a *dyn_f* value of 14. The bits are packed tightly, by concatenating all of the horizontal raster rows into one long string, and then packing this string eight bits to a byte. The number of bytes required can be calculated by $(width*height+7)$ **div** 8. This format should only be used when packing the character by run counts takes more bytes than this, although, of course, it is legal for any character. Any extra bits in the last byte should be set to zero.

12. At this point, we are ready to introduce the format for a character descriptor. It consists of

three parts: a flag byte, a character preamble, and the raster data. The most significant four nybbles of the flag byte yield the *dyn_f* value for that character. (Notice that only values of 0 through 14 are legal for *dyn_f*, with 14 indicating a bit mapped character; thus, the flag bytes do not conflict with the command bytes, whose upper nybble is always 15.) The next bit (with weight 16) indicates whether the first run count is a black count or a white count, with a one indicating a black count. For bit-mapped characters, this bit should be set to a zero. The next bit (with weight 8) indicates whether certain later parameters (referred to as size parameters) are given in one-byte or two-byte quantities, with a one indicating that they are in two-byte quantities. The last two bits are concatenated on to the beginning of the length parameter in the character preamble, which will be explained below.

However, if the last three bits of the flag byte are all set (normally indicating that the size parameters are two-byte values and that a 3 should be prepended to the length parameter), then a long format of the character preamble should be used instead of one of the short forms.

Therefore, there are three formats for the character preamble, and which one is used depends on the least significant three bits of the flag byte. If the least significant three bits are in the range zero through three, the short format is used. If they are in the range four through six, the extended short format is used. Otherwise, if the least significant bits are all set, then the long form of the character preamble is used. The preamble formats are explained below.

Short form: *flag*[1] *pl*[1] *cc*[1] *tfm*[3] *dm*[1] *w*[1] *h*[1] *hoff*[+1] *voff*[+1]. If this format of the character preamble is used, the above parameters must all fit in the indicated number of bytes, signed or unsigned as indicated. Almost all of the standard TEX font characters fit; the few exceptions are huge fonts such as `aminch`.

Extended short form: *flag*[1] *pl*[2] *cc*[1] *tfm*[3] *dm*[2] *w*[2] *h*[2] *hoff*[+2] *voff*[+2]. Larger characters use this extended format.

Long form: *flag*[1] *pl*[4] *cc*[4] *tfm*[4] *dx*[4] *dy*[4] *w*[4] *h*[4] *hoff*[4] *voff*[4]. This is the general format which allows all of the parameters of the GF file format, including vertical escapement.

The *flag* parameter is the flag byte. The parameter *pl* (packet length) contains the offset of the byte following this character descriptor, with respect to the beginning of the *tfm* width parameter. This is given so a PK reading program can, once it has read the flag byte, packet length, and character code (*cc*), skip over the character by simply reading this many more bytes. For the two short forms of the character preamble, the last two bits of the flag byte should be considered the two most-significant bits of the packet length. For the short format, the true packet length might be calculated as $(flag \bmod 4) * 256 + pl$; for the extended format, it might be calculated as $(flag \bmod 4) * 65536 + pl$.

The *w* parameter is the width and the *h* parameter is the height in pixels of the minimum bounding box. The *dx* and *dy* parameters are the horizontal and vertical escapements, respectively. In the short formats, *dy* is assumed to be zero and *dm* is *dy* but in pixels; in the long format, *dx* and *dy* are both in pixels multiplied by 2^{16}. The *hoff* is the horizontal offset from the upper left pixel to the reference pixel; the *voff* is the vertical offset. They are both given in pixels, with right and down being positive. The reference pixel is the pixel which occupies the unit square in METAFONT; the METAFONT reference point is the lower left hand corner of this pixel. (See the example below.)

13. TEX requires that all characters which have the same character codes modulo 256 also have the same *tfm* widths, and escapement values. The PK format does not itself make this a requirement, but in order for the font to work correctly with the TEX software, this constraint should be observed. The current version of TEX (1.5) cannot output character codes greater than 255 anyway.

Following the character preamble is the raster information for the character, packed by run counts or by bits, as indicated by the flag byte. If the character is packed by run counts and the required number of nybbles is odd, then the last byte of the raster description should have a zero for its least significant nybble.

14. As an illustration of the PK format, the character Ξ from the font amr10 at 300 dots per inch will be encoded. This character was chosen because it illustrates some of the borderline cases. The raster for the character looks like this (the row numbers are chosen for convenience, and are not METAFONT's row numbers.)

```
 0      MMMMMMMMMMMMMMMMMMMM
 1      MMMMMMMMMMMMMMMMMMMM
 2      MMMMMMMMMMMMMMMMMMMM
 3      MMMMMMMMMMMMMMMMMMMM
 4      MM                MM
 5      MM                MM
 6      MM                MM
 7
 8
 9          MM            MM
10          MM            MM
11          MM            MM
12          MMMMMMMMMMMMMMMMMM
13          MMMMMMMMMMMMMMMMMM
14          MMMMMMMMMMMMMMMMMM
15          MMMMMMMMMMMMMMMMMM
16          MM            MM
17          MM            MM
18          MM            MM
19
20
21
22     MM                    MM
23     MM                    MM
24     MM                    MM
25     MMMMMMMMMMMMMMMMMMMMMM
26     MMMMMMMMMMMMMMMMMMMMMM
27     MMMMMMMMMMMMMMMMMMMMMM
28  *  MMMMMMMMMMMMMMMMMMMMMM
```

The width of the minimum bounding box for this character is 20; its height is 29. The '*' represents the reference pixel; notice how it lies outside the minimum bounding box. The *hoff* value is -2, and the *voff* is 28.

The first task is to calculate the run counts and repeat counts. The repeat counts are placed at the first transition (black to white or white to black) in a row, and are enclosed in brackets. White counts are enclosed in parentheses. It is relatively easy to generate the counts list:

```
82 [2] (16) 2 (42) [2] 2 (12) 2 (4) [3]
16 (4) [2] 2 (12) 2 (62) [2] 2 (16) 82
```

Note that any duplicated rows that are not all white or all black are removed before the repeat counts are calculated. The rows thus removed are rows 5, 6, 10, 11, 13, 14, 15, 17, 18, 23, and 24.

15. The next step in the encoding of this character is to calculate the optimal value of *dyn_f*. The details of how this calculation is done are not important here; suffice it to say that there is a simple algorithm which in one pass over the count list can determine the best value of *dyn_f*. For this character, the optimal value turns out to be 8 (atypically low). Thus, all count values less than or equal to 8 are packed in one nybble; those from nine to $(13-8)*16+8$ or 88 are packed in two nybbles. The run encoded values now become (in hex, separated according to the above list):

```
D9 E2 97 2 B1 E2 2 93 2 4 E3
97 4 E2 2 93 2 C5 E2 2 97 D9
```

which comes to 36 nybbles, or 18 bytes. This is shorter than the 73 bytes required for the bit map, so we use the run count packing.

16. The short form of the character preamble is used because all of the parameters fit in their respective lengths. The packet length is therefore 18 bytes for the raster, plus eight bytes for the character preamble parameters following the character code, or 26. The *tfm* width for this character is 640796, or 9C71C in hexadecimal. The horizontal escapement is 25 pixels. The flag byte is 88 hex, indicating the short preamble, the black first count, and the *dyn_f* value of 8. The final total character packet, in hexadecimal, is:

Flag byte	88
Packet length	1A
Character code	04
tfm width	09 C7 1C
Horizontal escapement (pixels)	19
Width of bit map	14
Height of bit map	1D
Horizontal offset (signed)	FE
Vertical offset	1C
Raster data	D9 E2 97
	2B 1E 22
	93 24 E3
	97 4E 22
	93 2C 5E
	22 97 D9

Tomas Rokicki
Rokicki@SU-Score.Arpa
Department of Computer Science
Stanford University
Stanford, CA 94305

Hyphenation Exception Log

Below is a list of words that TEX fails to hyphenate properly. It last appeared in Volume 5, No. 1, on page 15. Everything listed there is repeated here.

As expected, there are more instances of missed than of incorrect hyphens. The first column gives results from TEX's `\showhyphens{...}`; entries in the second column are suitable for inclusion in a `\hyphenation{...}` list.

In most instances, inflected forms are not shown for nouns and verbs; however, all forms should be specified in a `\hyphenation{...}` list if they might occur in your document.

anal-yse	an-a-lyse
anomaly	anom-aly
anti-nomy(ies)	an-tin-o-my(ies)
ap-pendix	ap-pen-dix
ban-dleader	band-leader
be-haviour	be-hav-iour
bornolog-i-cal	bor-no-log-i-cal
Brow-n-ian	Brown-ian
buz-zword	buzz-word
cartwheel	cart-wheel
cholesteric	cho-les-teric
database	data-base
dat-a-p-ath	data-path
de-mos	demos
dis-tribute	dis-trib-ute
Di-jk-stra	Dijk-stra
elec-trome-chan-i-cal	electro-mechan-i-cal
elec-tromechanoa-cous-tic	electro-mechano-acoustic
equiv-ari-ant	equi-vari-ant
Eu-le-rian	Euler-ian
fermions	fermi-ons
flowchart	flow-chart
Gaus-sian	Gauss-ian
ge-o-met-ric	geo-met-ric
Greif-swald	Greifs-wald
Grothendieck	Grothen-dieck
Grundlehren	Grund-leh-ren
Hamil-to-nian	Hamil-ton-ian
Her-mi-tian	Her-mit-ian
hex-adec-i-mal	hexa-dec-i-mal
in-fras-truc-ture	in-fra-struc-ture
jeremi-ads	je-re-mi-ads
Kadomt-sev	Kad-om-tsev
Leg-en-dre	Le-gendre
Le-ices-ter	Leices-ter
Lip-s-chitz(ian)	Lip-schitz(-ian)
macroe-co-nomics	macro-eco-nomics
manuscript	man-u-script
Marko-vian	Markov-ian
met-a-lan-guage	meta-lan-guage
mi-croe-co-nomics	micro-eco-nomics
mi-crofiche	mi-cro-fiche
mod-elling	mod-el-ling
mo-noen-er-getic	mono-en-er-getic
monopole	mono-pole
monos-trofic	mono-strofic
mul-ti-pli-ca-ble	mul-ti-plic-able
ne-ofields	neo-fields
Noethe-rian	Noether-ian
none-mer-gency	non-emer-gency
nonequiv-ari-ance	non-equi-vari-ance
noneu-clidean	non-euclid-ean
non-s-mooth	non-smooth
parametrized	pa-ram-e-trized
paramil-i-tary	para-mil-i-tary
Poincare	Poin-care
polyene	poly-ene
poly-go-niza-tion	polyg-on-i-za-tion
postam-ble	post-am-ble
pream-ble	pre-am-ble
pseu-dod-if-fer-en-tial	pseu-do-dif-fer-en-tial
pseud-ofi-nite	pseu-do-fi-nite
pseud-ofinitely	pseu-do-fi-nite-ly
pseud-o-forces	pseu-do-forces
pseu-doword	pseu-do-word
quadrat-ics	qua-drat-ics
quasiequiv-a-lence	qua-si-equiv-a-lence
quasi-hy-ponor-mal	qua-si-hy-po-nor-mal
qua-sis-mooth	qua-si-smooth
qua-sis-ta-tion-ary	qua-si-sta-tion-ary
Rie-man-nian	Rie-mann-ian
schedul-ing	sched-ul-ing
Schrodinger	Schro-ding-er
Schwarzschild	Schwarz-schild
semidef-i-nite	semi-def-in-ite
semi-ho-mo-th-etic	semi-ho-mo-thet-ies
ser-vomech-a-nism	ser-vo-mech-anism
setup	set-up
solenoid	so-le-noid
spheroid	spher-oid
stochas-tic	sto-chas-tic
sub-scriber	sub-scrib-er
summable	sum-ma-ble
ther-moe-las-tic	ther-mo-elas-tic
times-tamp	time-stamp
ve-r-all-ge-mein-erte	ver-all-ge-mein-erte
wahrschein-lichkeit-s-the-o-rie	Wahr-schein-lich-keits-the-o-rie
waveg-uide	wave-guide

Beginning with this issue, "Font Forum" will be a regular feature in *TUGBoat*. Now, the name was not chosen simply for the pleasant alliteration, or for the fact that a Latinism like 'forum' would look nice in chiseled majuscules, but because it is intended to both spark and showcase open and hopefully spirited discussion of the thorny questions of typefaces as they relate to the use of TEX. The author has noticed an unsettling lack of awareness among the TEX community at large of the absolutely vital importance of the availability of many and beautiful typefaces to the continued growth and vigor of TEX.

Naturally, each of us interested in the issues related to typefaces and TEX has a peculiar background and certain favorite ways of doing things; and very few column inches will elapse before the author reveals herself in her true colors, as a confirmed devotee of METAFONT, on the subject of which we are about to be treated to a brief panegyric. Designing with METAFONT is not the easiest of techniques to master, for a number of reasons; and the reason that I wish to mention here is one of METAFONT's chiefest strengths, namely, that it gives the designer control over the letter-form at a *profound* rather than a *superficial* level. I do not at the moment wish to dwell on the implications of that fact for the process of design, but for the *product* of design. In sum, good METAFONT design results in typefaces that are unsurpassed in quality. A derived design (by which I mean a replica of a given typeface in a digital medium) done with METAFONT is truer to the original than one done with any other method I know of; METAFONT allows consideration of the niceties of stem and serif growth, and the tiny variations in letter proportion that are lost with techniques that involve scaling. And any design, whether derived or original, if done with METAFONT, has the very considerable advantage of being able to be easily, quickly and properly tuned to various output devices. Because of this, a document can be produced in every stage from roughest draft to final copy in the same typeface. Finally, it is worth reminding the reader that, when we create a METAFONT font, we have really created two very different but equally important files: one, the digital image for the output device, and the other, the TEX Font Metric (.tfm) file. The point here is that the .tfm file is not merely an after-the-fact addendum to a METAFONT font; it is created taking into account all the details in letter-form variation that I spoke of a moment ago.

The discussion of METAFONT, then, will form a substantial part of the content of this column. The all-new, all-improved METAFONT is about to become available; I hope to see the results of experiments with it soon. All users of METAFONT are therefore encouraged to send samples of the results of such experiments to share them with the readership of *TUGBoat*. Even unexpected or unwanted results can be very illuminating, as Don Knuth demonstrates in his set of 'meta-flops' ('Lessons Learned from METAFONT,' *Visible Language*).

Of course, publication of one's wanderings along the road to digital correctness will elicit remarks on the Wanderer's apparent progress; and 'Font Forum' will serve as a sounding board for such thoughtful criticism. The author is less disturbed (intellectually, at least) by round and hearty disapproval of her work than by acceptance of it by a user who feels unqualified to make any comment. The latter attitude is as illogical as saying, "This egg is rotten, but it is a better egg than I can lay; therefore, I shall eat it and like it". The former, of course, is unhelpful only if carried to unreasonable extremes, such as failing to consider the strictures of the medium in assessing the product. The upshot of this is simply that all thoughtful criticism of any font that is intended for use with TEX is fair grist for the 'Font Forum'.

However, 'Font Forum' is not intended primarily for users of METAFONT, but for users of TEX who happen to also be users of fonts. Some topics that I hope to see some discussion on in future issues of *TUGBoat* include: .gf format; issues related to the appro-

priate use of fonts; POEMS (a font format converter); development of non-METAFONT fonts for use with TEX (several shops, including Adobe, are presently deriving .tfm files for this purpose); general considerations on the process of digital font design and creation; even 'wish-lists' of character sets and/or type styles are appropriate.

Any and all such contributions should be sent to the author's address given at the end of this column.

New Developments with METAFONT

As I mentioned above, a new and improved version of METAFONT is about to be released; as of this writing, Version 0.95 is available. Don Knuth is preparing a complete guide to its use, *The Metafontbook*, and has graciously been making prepublication copies of it available through his office. The first print run was exhausted this summer, though, and Dr. Knuth was estimating that a new issue would not be made until sometime in October. I taught a course intended to serve as an introduction to METAFONT, 'The Elements of METAFONT Style,' at this summer's TUG meeting. This two-day seminar was enlivened by the perceptive remarks and questions of the brave souls who enrolled, and enriched by the contributions of Neenie Billawala, John Hobby, Don Knuth and Richard Southall, who shared some of their experience with METAFONT with the class. A videotape of the seminar is available from TUG.

An important point made again and again in the course deserves one more iteration here: and that is the fact that METAFONT has not been used as much as it deserves to be. To quote from the course manual:

> . . .METAFONT is a tremendously subtle and precise tool and *any* competent use of it requires a blend of two disparate mentalities, the analytic and the æsthetic. One lesson that the author hopes the reader has learned from this course is that creating an entire true meta-font is many times again harder than creating two or three meta-characters; and that creating a meta-family is many, many times harder than that. In fact, to achieve this latter requires yet another combination of disparate mentalities: one must be recklessly foolhardy to dive into such a task in the first place, and one must be doggedly level-headed to bring it to completion.
>
> But while meta-font design may be a task for only a handful, and meta-family design is a task for fewer even than that, some use of METAFONT *is* for everyone. It is eminently useful for such tasks as creating logos, specialized character sets, dingbats, borders and so on. Such tasks are of less monumental scale than font or family design, but of no less importance to the final appearance of a document.

Suggested Reading

Knuth, Donald E., "The Concept of a Meta-Font," pp. 3-2, *Visual Language*, Volume XVI, Number 1, Winter 1982

Knuth, Donald E., "Lessons Learned from Metafont," pp. 35-53, *Visual Language*, Volume XIX, Number 1, Winter 1985

Georgia K.M. Tobin
The Metafoundry
OCLC, Inc., MC 485
6565 Frantz Road
Dublin, Ohio 43017

Every character used to typeset 'Font Forum' was designed by G.K.M. Tobin using METAFONT. Printing was done on a Canon LBP-CX with a resolution of 300 dpi.

Mathematical Symbols and Cyrillic Fonts Ready for Distribution (Revised)

Barbara Beeton
American Mathematical Society

The first general release of fonts created at the American Mathematical Society was in August, at the time of the TUG meeting. This first release consists of cyrillic and two 128-character fonts of mathematical symbols, all in various sizes and weights. It is our intention that these fonts be added to the standard distribution, and we will make an effort to provide the necessary files to all sites from which the TEX package is being distributed. It will not be possible to provide this material directly to users, since the Society's DEC 20 computer has proved singularly unsuitable for making tapes that can be read by any other kind of machine.

A master tape was delivered to Stanford and installed on the Score (DEC-20) system, in the directory `<TeX.AMSFONTS>`. This directory now contains the METAFONT78 sources necessary to generate the fonts in distribution format, as well as corresponding `.PXL` files, along with several files of macros, documentation and user instructions. Actual addition of this material to distribution tapes (and PC diskettes) is now being arranged; information regarding the fonts was sent to all principal distribution sites, and arrangements are being made to transmit the necessary files as soon as practicable.

Readers of the original article in the last TUGboat will note that the font names presented there have been changed. (The content of the symbol fonts has also changed slightly, to omit redundancies.) We have been persuaded that the name "Euler" should be reserved for the professionally-designed fonts commissioned by the Society from Hermann Zapf — little enough recognition is given even the best font designers, without the font names associated with their work being wrongfully attributed to something that they had no hand in creating. We acknowledge our error, and hope our apology is accepted.

	′0	′1	′2	′3	′4	′5	′6	′7	
′00x	Њ	Љ	Ц	Э	І	Є	Ъ	Ћ	″0x
′01x	њ	љ	ц	э	і	є	ђ	ћ	
′02x	Ю	Ж					S	Я	″1x
′03x	ю	ж					s	я	
′04x	¨	!	”		˜	%	´	’	″2x
′05x	(	)	*		,	-	.	/	
′06x	0	1	2	3	4	5	6	7	″3x
′07x	8	9	:	;	«	ı	»	?	
′10x		А	Б	Ц	Д	Е	Ф	Г	″4x
′11x	Х	И	Й	К	Л	М	Н	О	
′12x	П	Ч	Р	С	Т	У	В	Щ	″5x
′13x	Ш	Ы	З	[	“	]	Ь	Ъ	
′14x	‘	а	б	ц	д	е	ф	г	″6x
′15x	х	и	й	к	л	м	н	о	
′16x	п	ч	р	с	т	у	в	щ	″7x
′17x	ш	ы	з	–	—	№	ь	ъ	
	″8	″9	″A	″B	″C	″D	″E	″F	

The AMS cyrillic font – MCYR10

Мещанский университет, находящийся на пути к Нескучному, праздновал на днях свой пятидесятилетний юблией. Кого возили в Титы или городскую больницу, тот, конечно, помнит здороввеннейший, трёхэтажный домище по правую руку с вывеской «Богадельная и Мещанские училища» и тому наверное встречались на пути вереницы ученических пар, солидно прогуливаемых надзирателями.

Cyrillic

The cyrillic font contains all letters found in the modern (post-revolutionary) cyrillic alphabet, as well as others found by Mathematical Reviews to be necessary for rendering bibliographic information in Russian, Ukrainian, Serbian, Georgian, and other Slavic and non-Slavic languages ordinarily published in cyrillic. Accents which normally occur in these languages, as well as in such words as names of mathematicians whose work is regularly translated into Russian or one of the other languages covered by MR, are included in the font, as are all the digits and ordinary punctuation. Several cells are still empty: the number of such cells is not sufficient to hold all the additional pre-revolutionary Russian letters, and there is not yet enough experience to indicate what else might most usefully (for MR) be included.

The "basic" cyrillic font is `MCYR10`. Names have been assigned to a number of variations, not all of which exist yet. (In particular, there are no plans yet to create the METAFONT descriptions of the true "italic" letters.)

`MCYR`	lightface	`MCSL`	slanted
`MCB`	bold	`MCBSL`	bold slanted
`MCBX`	bold extended	`MCSS`	sans serif
`MCI`	italic	`MCSSB`	bold sans serif
`MCBI`	bold italic		

Font names have been assigned so that compression to 6 characters, using the first 3 and last 3 letters of longer names (a standard built into most implementations of TEX for operating systems having such a limit, and announced through TEXhax by David Fuchs), will always be unique.

Keying of cyrillic to be rendered with this font is in accord with the current MR transliteration scheme, e.g.

Khrushchëv (`Khrushch\"ev`) → Хрущёв
Zhurnal (`Zhurnal`) → Журнал
Kiïv (`Ki{\"\i}v`) → Київ

`\font\tencyr=mcyr10 \def\cyr{\tencyr\cyracc}` `{\cyr ...}` sets the stage for proper transition to and from cyrillic. Most of the translation from keyed input to cyrillic is implemented by ligature instructions in the font itself. A few letters require a "chain" of ligatures: sh → ш, shc → ч (an obvious absurdity, but the roman combination never occurs legitimately), shch → щ. Letters rendered with accents in transliteration are trapped by macros defined in the file `CYRACC.DEF`: й (`\u\i`) → й, ī (`\=\i`) → i, ï (`\"\i`) → ï. Both the macro accent traps and ligatures do the right thing in ordinary roman text and in `{\cyr ...}`; however, cyrillic items in tables cannot reliably be specified in the preamble, and in individual cells `\cyr` should be preceded by `\relax` to prevent premature expansion, and thus loss, of the macro instructions.

Hyphenation is not automatically suppressed, but the patterns used will be those for English in the absence of a local override. (We do not know whether any Russian patterns exist.) For short passages, as the above sample, or isolated words, good luck may prevail.

Documentation accompanying this font will include full keying instructions, the ligature specifications, and, of course, `CYRACC.DEF`. Should the transliteration scheme in local use be different from the MR scheme (for example, an earlier MR scheme rendered щ as šč), it should be quite easy to modify `CYRACC.DEF` to accommodate it, and, if ligature changes are absolutely necessary, they may be implemented using the TEXware programs `TFtoPL` and `PLtoTF`.

Mathematical symbols

Mathematicians expanding the boundaries of their chosen areas often find that no suitably unambiguous notation exists with which to express new concepts. First attempts usually consist in seeking out ever more exotic alphabets, but this fount is rather rapidly exhausted. Non-alphabetic symbols modeled after, or constructed from combinations of, existing ones is probably the next most profitable approach. And failure in either of those attempts may yield something truly new. In any event, the net result is proliferation of symbols beyond what is available to most ordinary typesetting systems.

The original symbol fonts, `CMSY` and `CMEX` (currently `AMSY` and `AMEX`), contain the most frequently used mathematical symbols, plus whatever else was needed for *The Art of Computer Programming*, volume 2, and other projects that Don Knuth was working on at the time. Many other symbols are in common use in other subfields of mathematics, and the AMS and MR found it necessary to construct them. We have now filled one entire "extra symbols" font and most of a second.

The naming scheme devised for these "extra symbols" fonts also leaves room for a third. "Medium" and "bold" refer to the weight, medium being matched to the weight of the "basic" Computer Modern symbols in the `CMSY` font.

`MSXM`	symbols 1 medium	`MSXB`	symbols 1 bold
`MSYM`	symbols 2 medium	`MSYB`	symbols 2 bold
`MSZM`	symbols 3 medium	`MSZB`	symbols 3 bold

Negated relations.

2 "04	$\not<$	\nless	2 "05	$\not>$	\ngtr	2 "1C	$\not\sim$	\nsim
2 "02	$\not\leq$	\nleq	2 "03	$\not\geq$	\ngeq	2 "1D	$\not\approx$	\napprox
2 "0A	$\not\leqslant$	\nleqslant	2 "0B	$\not\geqslant$	\ngeqslant	2 "2E	$\nmid$	\nshortmid
2 "14	$\not\leqq$	\nleqq	2 "15	$\not\geqq$	\ngeqq	2 "2F	$\nparallel$	\nshortparallel
2 "0C	$\lneq$	\lneq	2 "0D	$\gneq$	\gneq	2 "2D	$\nmid$	\nmid
2 "08	$\lneqq$	\lneqq	2 "09	$\gneqq$	\gneqq	2 "2C	$\nparallel$	\nparallel
2 "00	$\lvertneqq$	\lvertneqq	2 "01	$\gvertneqq$	\gvertneqq	2 "30	$\nvdash$	\nvdash
2 "12	$\lnsim$	\lnsim	2 "13	$\gnsim$	\gnsim	2 "32	$\nvDash$	\nvDash
2 "1A	$\lnapprox$	\lnapprox	2 "1B	$\gnapprox$	\gnapprox	2 "31	$\nVdash$	\nVdash
2 "06	$\nprec$	\nprec	2 "07	$\nsucc$	\nsucc	2 "33	$\nVDash$	\nVDash
2 "0E	$\npreceq$	\npreceq	2 "0F	$\nsucceq$	\nsucceq	2 "36	$\ntriangleleft$	\ntriangleleft
2 "16	$\precneqq$	\precneqq	2 "17	$\succneqq$	\succneqq	2 "37	$\ntriangleright$	\ntriangleright
2 "10	$\precnsim$	\precnsim	2 "11	$\succnsim$	\succnsim	2 "35	$\ntrianglelefteq$	\ntrianglelefteq
2 "18	$\precnapprox$	\precnapprox	2 "19	$\succnapprox$	\succnapprox	2 "34	$\ntrianglerighteq$	\ntrianglerighteq
2 "2A	$\nsubseteq$	\nsubseteq	2 "2B	$\nsupseteq$	\nsupseteq			
2 "22	$\nsubseteqq$	\nsubseteqq	2 "23	$\nsupseteqq$	\nsupseteqq			
2 "28	$\subsetneq$	\subsetneq	2 "29	$\supsetneq$	\supsetneq			
2 "20	$\varsubsetneq$	var. \subsetneq	2 "21	$\varsupsetneq$	var. \supsetneq			
2 "24	$\subsetneqq$	\subsetneqq	2 "25	$\supsetneqq$	\supsetneqq			
2 "26	$\varsubsetneqq$	var. \subsetneqq	2 "27	$\varsupsetneqq$	var. \subsetneqq			

Arrows.

1 "12	$\leftleftarrows$	\leftleftarrows	1 "13	$\rightrightarrows$	\rightrightarrows	1 "14	$\upuparrows$	\upuparrows
1 "1C	$\leftrightarrows$	\leftrightarrows	1 "1D	$\rightleftarrows$	\rightleftarrows	1 "15	$\downdownarrows$	\downdownarrows
1 "57	$\Lleftarrow$	\Lleftarrow	1 "56	$\Rightarrow$	\Rrightarrow	1 "0A	$\updownarrow$	\updownarrow
1 "11	$\twoheadleftarrow$	\twoheadleftarrow	1 "10	$\twoheadrightarrow$	\twoheadrightarrow	1 "18	$\upharpoonleft$	\upharpoonleft
1 "1B	$\leftarrowtail$	\leftarrowtail	1 "1A	$\rightarrowtail$	\rightarrowtail	1 "19	$\downharpoonleft$	\downharpoonleft
1 "22	$\looparrowleft$	\looparrowleft	1 "23	$\looparrowright$	\looparrowright	1 "16	$\upharpoonright$	\upharpoonright
2 "78	$\curvearrowleft$	\curvearrowleft	2 "79	$\curvearrowright$	\curvearrowright	1 "17	$\downharpoonright$	\downharpoonright
1 "09	$\circlearrowleft$	\circlearrowleft	1 "08	$\circlearrowright$	\circlearrowright	1 "0B	$\leftrightharpoons$	\leftrightharpoons
1 "1E	$\Lsh$	\Lsh	1 "1F	$\Rsh$	\Rsh	1 "28	$\multimap$	\multimap
1 "20	$\rightsquigarrow$	\rightsquigarrow	1 "21	$\leftrightsquigarrow$	\leftrightsquigarrow			

"Negated" arrows.

2 "38	$\nleftarrow$	\nleftarrow	2 "39	$\nrightarrow$	\nrightarrow	2 "3D	$\nleftrightarrow$	\nleftrightarrow
2 "3A	$\nLeftarrow$	\nLeftarrow	2 "3B	$\nRightarrow$	\nRightarrow	2 "3C	$\nLeftrightarrow$	\nLeftrightarrow

Delimiters.

1 "70	$\ulcorner$	\ulcorner	1 "71	$\urcorner$	\urcorner
1 "78	$\llcorner$	\llcorner	1 "79	$\lrcorner$	\lrcorner

Non-math symbols.

1 "58	✓	\checkmark	1 "72	®	\circledR
1 "7A	✠	\maltese	1 "55	¥	\yen

Alternate names.

1 "6E	$\lll$	\llless	1 "6F	$\ggg$	\gggtr	1 "2B	$\Doteq$	\Doteq
1 "65	$\Cap$	\doublecap	1 "64	$\Cup$	\doublecup	1 "16	$\restriction$	\restriction

Output Devices

Output Devices and Computers

Table I-a: Proof-quality devices attached to "large" computers and workstations

	Amdahl (MTS)	Apollo	CDC Cyber	DEC 10	DEC 20	DG MV	Ether-net	HP9000 Ser.500	IBM (MVS)	IBM (VM)	PERQ	Prime	Siemens (BS2000)	Sun	VAX (Unix)	VAX (VMS)
Apple LaserWriter		Textset†		Textset	Textset†			Textset†	Textset	Textset				Textset†	Crltn; Textset†	Textset†
C Itoh																LSU
Canon									GMD		GMD		GMD		Canon	
DEC LN01															UWash	LSU
DEC Ltr Ptr 100					OSU[d]											
DEC VT125																INFN
Diablo												OSU[p]				
Facit 4542																INFN
Fla Data					MR											
GE 3000		COS														
HP 2680							Stnfd									
IBM 3800; 4250;Sherpa										SLAC						
Imagen	UBC; Textset	OCLC; Textset†		Stnfd; Vndblt; Textset†	SRI; Clmbia; Textset†	TAMU	Imagen		Textset	SLAC; Textset				Sun; Textset†	UMd; Textset	K&S†; Textset
NDK 7700										IAM						
PostScript printers	Textset	Textset		Textset	Textset			Textset	Textset	Textset				Textset	Textset	Textset
Printronix						TAMU										
QMS Lasergrafix	Textset	ScnLsr; Textset		Textset	Textset	TAMU		TAMU	Textset	Textset	GMD	TAMU		Textset	Textset; UWash	TAMU; Textset
screen prevue		Yale; Textset				TAMU			GMD		GMD		GMD	Textset; UCB		Adld
Symbolics					UWash										UWash	Calma
Talaris				Talrs†	Talrs†				Talrs†	WashStU					Talrs†	Talrs†
Tektronix 4014									UMilan							Adld; INFN
Toshiba 135x						TAMU										
Varian					AMS											SciAp
Versatec			UKöln	GATch; Vndblt	UWash	TAMU			UMilan	Wzmn		Lvmr			UWash	K&S†
Xerox Dover					CMU		Stnfd								Stnfd	
Xerox 2700			Bochum													
Xerox 2700II					OSU[d]											
Xerox 9700	UMich; Textset	COS; Textset		UDel					Textset	UDel; Textset				Textset	Textset	ACC; Textset

Notes:

* Still running TEX80

† Graphics supported

‡ Computer used only to support output device, not to run TEX at this installation.

Table I-b: Proof-quality devices attached to "small" computers						
	Apple Macintosh	HP 1000	HP 3000	HP 9000 Ser. 200	IBM PC	TI PC
Apple ImageWriter	K&S†					
Apple LaserWriter	K&S†			Textset	PCTEX; Textset	
Corona Laser Printer 300					PCTEX	
Diablo			TEXET			
Epson		JDJW			A-W; PCTEX	TAMU†
HP 2680			TEXET			
HP 2688A				HP; CaTch		
Imagen					OCLC‡; PCTEX; Textset	
Printronix						TAMU†
QMS Lasergrafix					PCTEX; Textset	
Qume			TEXET			
screen prevue	K&S†				PCTEX; Textset†	TAMU†
TI 855						TAMU†
Toshiba					PCTEX	

Table II: Typesetters													
	Amdahl (MTS)	Apollo	CDC Cyber	DEC20	HP3000	HP9000	IBM (MVS)	IBM (VM)	IBM PC	Sun	Univac 1100	VAX (Unix)	VAX (VMS)
Agfa P400								IAM					
Alphatype CRS				AMS									
Autologic APS-5/Micro-5	Textset	COS; Textset		Textset	Textset	HP	Textset	Textset	PCTEX	Textset		Textset	Intergraph†; Textset
Compugraphic 8400					USheffield				PCTEX				K&S
Compugraphic 8600			RECAU*				WashStU	WashStU	PCTEX		UWis*		K&S
CRTronic													Eire
Harris 7500												SARA	
Linotron 202				Adapt					PCTEX				

Most of the interfaces listed in these charts are not on the standard distribution tapes. Some are considered proprietary. Information regarding these interfaces should be obtained directly from the sites listed.

Output device data is being maintained by Rilla Thedford. Anyone desiring more information or relaying new information can send it to her at the address given on the reverse of the title page or via the Arpanet:

`Rilla_Thedford%UMich-MTS@MIT`

The codes used in the charts are interpreted below, with a person's name given for a site when that information could be obtained and verified. If a contact's name appears in the current TUG membership list, only a phone number or network address is given. If the contact is not a current TUG member, the most recent address, and its source, are shown.

ACC (Advanced Computer Communications): Diane Cast, 720 Santa Barbara St., Santa Barbara, CA 93101, 805-963-9431 (DECUS, May '85)
Adapt (Adapt, Inc): Marc Berkowitz, 415-393-9500
Adld (Adelaide University, Australia): Andrew Trevorrow, (08) 228 5984
AMS (American Math Society): Ron Whitney, 401-272-9500
A-W (Addison-Wesley): 617-944-3700, ext. 2677
Bochum (Ruhr Universität Bochum): Norbert Schwarz, 49 234 700-4014
Calma:
CaTch (Cal Tech): Glen Gribble, 818-356-6988
Canon (Tokyo): Masaaki Nagashima, (03)758-2111

Clmbia (Columbia): Frank da Cruz, 212-280-5126
CMU (Carnegie-Mellon University): Howard Gayle, 412-578-3042
COS (COS Information, Montreal): Kevin Small, 514-738-2191
Crltn (Carleton University): Neil Holtz, 613-231-7145
Eire (Bord Fáilte–Irish Tourist Board): James Cumiskey, Dublin 353-1-765871, ext. 1275
GATech (G A Technologies): Phil Andrews, 619-455-4583
GMD (Gesellschaft der Math und Datenfabrik, Bonn, Germany): Dr. Wolfgang Appelt
HP (Hewlett-Packard): Stuart Beatty, 303-226-3800, ext. 2067
IAM (Institut für Angewandte Math, Univ of Bonn, Germany): Bernd Schulze, 0228-733427
Imagen: Dan Curtis, 408-986-9400
INFN (INFN/CNAF, Bologna, Italy): Maria Luisa Luvisetto, 051-307572
Intgrph (Intergraph): Mike Cunningham, 205-772-2000
JDJW (JDJ Wordware): John D. Johnson, 415-965-3245
K&S (Kellerman & Smith): Barry Smith, 503-222-4234
LSU (Louisiana State University): Neal Stoltzfus, 504-388-1570
Lvmr (Lawrence Livermore Lab):
MR (Math Reviews): Dan Latterner, 313-996-5266
OCLC: Tom Hickey, 616-764-6075
OSU (Ohio State University): *DEC 20*: John Gourlay, 614-422-6653; *Prime*: John Crawford, 614-422-1741
PCTEX (Personal TEX, Inc.): Lance Carnes, 415-388-8853
RECAU (Aarhus University, Regional Computer Center):
SARA (Stichting Acad Rechenzentrum Amsterdam): Han Noot, Stichting Math Centrum, Tweede Boerhaavestraat 49, 1091 AL Amsterdam (TUGboat 5#1)
ScnLsr (Scan Laser, England): John Escott
SciAp (Science Applications): L. E. Fields, 619-458-2616
SLAC: Alan Spragens, 415-854-3300, ext. 2849
SRI:
Stnfd (Stanford):
Sun (Sun, Inc):
TAMU (Texas A&M): *HP 9000 Ser. 500*: Ken Marsh, 409-845-4940; *all others*: Bart Childs, 409-845-5470
TeXet: Lance Carnes, 415-388-8853
Textset (Ann Arbor, Mich.): Bruce Baker, 313-996-3566
Talrs (Talaris): Sonny Burkett, 619-587-0787
UBC (Univ of British Columbia): Afton Cayford, 604-228-3045
UCB (Univ of California, Berkeley): Michael Harrison, vortex@berkeley.arpa
UDel (Univ of Delaware): Daniel Grim, 302-451-1990
UKöln (Univ of Köln, Germany): Jochen Roderburg, 0221-/478-5372
UMd (Univ of Maryland): Chris Torek, 301-454-7690
UMich (Univ of Michigan): Hal Varian, 313-764-2364
UMilan (Università Degli Studi Milan, Italy): *Tektronix*: Dario Lucarella, 02/23.62.441 (329); *Versatec*: Giovanni Canzii, 02/23.52.93
USheffield (Univ of Sheffield, England): Ewart North, (0742)-78555, ext. 4307
UWash (Univ of Washington): Pierre MacKay, 206-543-2386
UWis (Univ of Wisconsin): William Kelly, 608-262-9501
Vndblt (Vanderbilt University): H. Denson Burnum, 615-322-2357
WashStU (Washington State University): Dean Guenther. 509-335-0411
Wzmn (Weizmann Institute, Rehovot, Israel): Malka Cymbalista, 08-482443
Yale: Bill Gropp, 203-436-3761

Index to Sample Output from Various Devices

Camera copy for the following items in this issue of TUGboat was prepared on the devices indicated, and can be taken as representative of the output produced by those devices. Some items (noted below) were received as copy larger than 100%; these were reduced photographically using the PMT process. The bulk of this issue, as usual, has been prepared (all with TEX82) on the DEC 2060 and Alphatype CRS at the American Mathematical Society.

- Apple LaserWriter (300 dpi): Textset advertisement, p. 163.
- Canon CX (300 dpi): Georgia Tobin, the Font Forum, p. 122.
- Corona LP300 (300dpi): PC TEX advertisement, p. 166.
- HITAC H8172 Laser Beam Printer (240 dpi): Tsunetoshi Hayashi, Report on TEX implementation at HUCC, p. 135; reduced from 125%; HITAC M280H.

Site Reports

The TEX Directories at the "Source"

For those TEX users who have access to the Arpanet, the most recent version of any file that will be put into the standard (generic) distribution can be found at SU-Score, the DEC-20 (TOPS-20) system that is "home" to the Stanford TEX Project. Here are the names of the directories to look in.

<TEX>
The latest executables can be found here.

<TEX.DOC>
All sorts of useful documentation — Start here with -READ-.ME and other -READ-.*.

<TEX.AMSFONTS>
.MF sources (old MF) and .PXL files for the cyrillic and extra symbol fonts developed at the American Mathematical Society.

<TEX.AMSTEX>
AMS-TEX (but not the *Joy of TEX*, at least not yet).

<TEX.BIBTEX>
The bibliographic system designed for use with LATEX.

<TEX.CM>
The Computer Modern fonts.

<TEX.FONTS>
Fonts — .TFM files and other representations.

<TEX.FORMATS>
.FMT files.

<TEX.INPUTS>
PLAIN.TeX et al.

<TEX.LATEX>
LATEX.

<TEX.MF>
METAFONT sources, in WEB.

<TEX.PXL>
.PXL files.

<TEX.SOURCES>
The TEX sources, in WEB.

<TEX.WEB>
WEB, Tangle and Weave.

TEX in the UK and Ireland: TEXline

Number 1 of the *TEXline*, the Newsletter of TEX users in UK and Ireland, appeared in August 1985. It begins with the following Editorial Note:

> This Newsletter is fairly experimental, and I hope, reasonably informal. I am happy to print any material which is sent to me for inclusion (or even to hand the whole thing to someone else who would like the joy of assembling it). Naturally any positive suggestions will be positively welcomed. I tend to see the general field of interest as quite wide, and for that reason have included material which is not directly TEX, but is TEX-like.

The content of the TEXline includes notices and reports of meetings, a summary of "useful(?) things from *texhax/laser lovers*", queries (nicely titled "Pleas, please"), an article on buying laser printers, a bibliography of books on TEX, and other interesting items. The next issue is scheduled to appear "by November". The editor/assembler of this welcome addition to the TEX literature is

Malcolm W. Clark
Imperial College Computer Center
Exhibition Road
London SW7 2BX, England

Keep up the good work, Malcolm.

Data General Distribution News

Bart Childs
Texas A & M University

The primary news items here are:

1. We have successfully ported TEX1.5. Of course, it was absolutely no problem. The previous change file was sufficient.
2. We have ported METAFONT 0.91. We will be porting 0.96 in a day or two.
3. We have learned how to make a proper nonVirTEX and are now making a generic routine for that.
4. Data General's PASCAL/VS is apparently one of the cleaner ones around and they have a new optimizer which makes another five or six percent improvement.

We will make a new distribution tape as soon as we have item 3 working, add graphic output to METAFONT, and have our drivers using the PK form of the new Computer Modern fonts. See Tom Rokicki's note about this (page 115).

Our driver for the QMS Lasergrafix printers has been adapted for the Imagen, a preview mechanism for some graphics terminals and other printers. As soon as we have the 'new fonts' working well as a standard, we will contribute its sources to the different distribution tapes. It correctly handles the differences for QMS' use of the Canon and Xerox marking engines, pages can be intermixed in portrait and landscape modes, random access of the DVI file is used to make documents for folding, two pages of output on one sheet for proofing folded documents, and downloaded and cached fonts are used.

We have three utility items which we will be happy to distribute on standard magnetic tapes to anybody. These are:

1. A braces checking program. Of course, this is not needed for EMACS users.
2. A RUNOFF to TEX converter. It was written in a rather brute force manner. We will be converting it to WEB.
3. A WORDSTAR to TEX converter. This was written in WEB.

MVS TEX Site Report

Craig Platt
University of Manitoba

At the recent TUG85 meeting, Alan Spragens and I agreed to share the rôle of IBM site coördinator, he for CMS and I for MVS and related systems. As mentioned by Alan in his article in the last issue, a CMS tape has been available for some time, but MVS hopefuls are still waiting. As a result of discussions with him and others at the meeting, I think we might see an MVS tape "Real Soon Now". I have been running a version 1.0 TEX (but with only limited device support), and have sent out my change files to a couple of other users who reported success at installing them. I am currently testing a 1.4 version which I think will serve for an initial distribution. With a few more sites on line, I hope suggestions for improvements will start appearing.

At the IBM birds-of-a-feather session, a couple of ideas were discussed which will enhance the package. The ever-present problem of ASCII vs. EBCDIC character encoding becomes particularly important for creation of a "load-and-go" version of TEX. TEX uses ASCII coding for its internal string processing, and converts these codes to system-dependent encodings through the *xchr* and *xord* arrays. If a site installs TEX by means of TANGLE and change files, these arrays can be adjusted to accomodate the local EBCDIC encoding—in fact this will be automatic if all text files on the distribution tape, including TANGLE.PAS, are character-translated first. But for a load-and-go version, a choice must be made at compile time. After some discussion about the ASCII-to-EBCDIC translation used by Stanford for the generic distribution tapes, we decided to try an idea from TEX80, namely the ASCII.TBL file. This file will be read at run-time by TEX (and by other programs that use internal ASCII encoding) and will be used to modify the *xchr* and *xord* tables appropriately. An installer will only have to edit this table to indicate any local exceptions from the "standard" translation. This scheme suffers from the disadvantage of requiring an extra file at run time, but it should be flexible enough to accomodate a variety of sites.

The tape will contain PLAIN.FMT, LATEX.FMT, and AMSTEX.FMT files, so these can be run directly (there is no plan to include a precompiled INITEX at the moment), but TEX source files needed by these packages (e.g., ".sty" files) will still have to be character-translated as they are unloaded from the tape.

Another point of discussion, relevant to MVS sites, was that of file naming conventions. In the earlier versions of TEX from Susan Plass and Eagle Berns, simple ddnames were used for file allocation. For a text file, the "area" and "extension" fields of a name were dropped and the remaining part used as a ddname for a sequential dataset. The current version builds on an idea suggested by Joey Tuttle last year, to add a bit of name parsing for more flexibility. Briefly, it makes up to three attempts to open an input file. First, a simple ddname is used as before. If that fails, a composite ddname is made by concatenating up to 5 characters of this simple name to the first three characters of the "extension". If this too is not found, then the extension alone is used as the ddname of a partitioned data set with the simple name as member (using the PDSIN option of PASCAL/VS). This last option will, for example, allow for libraries of macro files and other TEX source files which can be in partitioned data sets concatenated to the ddname TEX.

There is clearly more work to do here, and we really need a dynamic allocation capability, but that seems to be non-trivial in the MVS environment. If we can get a few more sites going with this first version, perhaps suggestions for improvements will result.

It was agreed that the "official" MVS and CMS distribution tapes ought to include the new METAFONT as well as the CMR sources, so that users can build their own GF (or PXL) files. This means the whole package probably won't be ready till near the year's end. People who just can't wait might want to contact me for an "as-is" tape of work in progress, which I could provide on an informal basis. By default, this would be an unlabelled 6250 bpi tape with a number of partitioned data sets, including the above-mentioned load modules

as well as change files and most of the contents of the standard distribution tape. This format is the one I would suggest for eventual distribution by the Codes. People should let me know if it would be unsuitable in any way, and we can try to provide options.

Finally, my mailing address is
Craig Platt
Department of Mathematics and Astronomy
University of Manitoba
Winnipeg, Manitoba R3T 2N2
Canada

Phone me at [204] 474-9832 during the day (with maximum probability of catching me between 1:30 and 2:30 CST on Mondays, Wednesdays, and Fridays). Otherwise you can leave a message at 474-8703 and I will return your call. Please indicate the best time to call back. If you have network access, try

```
platt%uofm-uts.cdn@ubc.csnet
```

or

```
platt%uofm-uts.cdn%ubc.csnet
                    @csnet-relay.arpa
```

TEX at WSU

Dean Guenther
Washington State University

At the TUG meeting, I briefly discussed and presented a video tape on our use of TEX and *TEXt1*, a macro package and on-line interface we have developed at Washington State University.

We have been using TEX heavily for the last three years. In a previous *TUGboat* (Vol. 5, No. 1) I mentioned some applications. Since then we have converted to TEX82, and continue generating many papers, reports, articles, etc. One of those was the *ACM-SIGUCCS 1984 Conference Proceedings* (ISBN 0-89791-146-6).

Other articles accepted as camera ready are *André Thevet on North America: A Sixteenth Century View* by Dr. Roger Scleshinger (Department of History) and Professor Arthur Stabler (Department of Foreign Languages), published by McGill-Queen's University Press. Another, on Big Bend National Park, is by Professor Jameson (History). Dr. Nicolas Kiessling (Department of English) is publishing the *Catalogue of Robert Burton's Library* with the Oxford Bibliographical Society. All three of these manuscripts are over 400 pages each. Dr. Bill Katra (Foreign Languages) has finished *Domingo F. Sarmiento: Public Writer Between 1839–1852*, published by the University of Arizona Press. Katra's *Sarmiento: Public Writer* and Kiessling's *Catalogue* were published using Computer Modern.

TEXt1 is a collection of macros and IBM VM/CMS based on-line interface. Whereas the macros are not unlike LaTEX, we felt we could add a little more flexibility with respect to tailored formats. What is significantly different is the interface, which gives menu selection of the main functions; create/modify, spell check, proofread, print, and a few others. There is also an extensive help facility for such things as how to use each of the commands, and how to use the menus. You can also send the output to a network of printers ranging from dot matrix printers, to laser printers, or to a typesetter, all available to each user on the system.

We started *TEXt1* in a pilot phase this last February with 40 faculty, staff, and graduate students. This fall we have moved to Beta testing, adding 250 more to the project, including undergraduates. We expect to be in full production this January.

Another item we worked on was getting IBM's Professional Office System (PROFS) to run under TEX instead of IBM's Document Composition Facility (DCF). We were successful in this venture, but decided it was not such a good idea after all. Not the least of reasons being that replacing DCF with TEX would render parts of the IBM supplied PROFS documentation useless and/or misleading. Also, PROFS supports other IBM products such as GDDM, HDDI (for Displaywriter integration), ISPF, etc. None of these were addressed in our test. Trying to keep a modified PROFS/TEX up to speed with IBM's current and upcoming products would create a task that we likely would never have the personnel to accomplish.

Some good news. It appears that Compugraphic has finally agreed to do something with METAFONT. Dave Fuchs has sent them the GF files for Computer Modern. They are presently testing the fonts, and expect to have the fonts available sometime next quarter. They will contact me with prices and availability. If you would like to know that information when it becomes available, call me (509-335-0411) or send me a note. I'm on BITNET, "GUENTHER AT WSUVM1".

Reports on TEX Implementation at HUCC

Hayashi, Tsunetoshi*

Basic Considerations

The TEX82 Version 1.0 transport at Hokkaido University Computing Center (HUCC) is almost completed this summer, 1985. The INITEX version TEX82 is currently running at HUCC, and I am intending to develop the production version as soon as possible. In the following, notes on decisions made in the attempt, and the measures devised to cope with problems found are presented. Note that this manuscript is typeset by TEX82, and printed by DVIwrite on HITAC H-8172 Laser Beam Printer.

The whole TEX82 and TEXware programs were founded on Pascal-H of DEC-20. This Pascal implementation deviates rather much from the Jensen-Wirth standard and TEX82 heavily relies on the extension. The amount of work required for transporting depends on how much the target Pascal compiler can support such extensions.

The target system is HITAC M280H/VOS3 at HUCC. HITAC M280H is a 16 MIPS 370 compatible machine; its performance is nearly equivalent to IBM model 3084. VOS3 is its operating system being "almost" compatible to MVS/TSO. The VOS3 Pascal provides none of above extensions, and conforms well to the standard except for external routine call and a few additional built-in procedures.

Default entry in case statement. This extensions is rather easily handled by WEB macro, syntax-oriented translation, or systematic source code editing. A syntax translation program which rewrites a **case** statement with default entry into an **if** statement and a **case** statement without default entry was written and used. In writing this program, I employed a parser generator software tool, which had been developed for compiler writing. It translates

```
case <expression> of
    n1: ...
    n2: ...
    ...
    otherwise <default statement>
end
```

into

```
if <expression> in [n1, n2, ...] then
    case <expression> of
        n1: ...
        n2: ...
        ...
    end
else <default statement>
```

This conversion would not keep the same meaning if the <expression> had had side effect when evaluated. Lucky enough, there is no such place in the TEX82 and TEXware programs. A remaining problem is that VOS3 Pascal can only accept a set expression where the domain of its integer or subrange element is between 0 and 2047. A set whose element is more than 2090 is generated by the above conversion. I had to hand-edit this part of the generated Pascal code.

Non-standard i/o. Extended i/o facilities such as dynamic file declaration, array of file, packed file, run-time file binding, interactive terminal i/o, and subtle terminal control are not provided by VOS3 Pascal. I employed proper record variables for defining file specifications and assembler-written i/o driver routines instead of Pascal file variables and i/o procedures. This solved both dynamic file declaration and array of file as well as look-ahead convention and run-time file binding.

Small integer packing. VOS3 Pascal can pack only quarter words into a word. This may cause increased storage requirement, nevertheless, this will not reduce run-time efficiency. TEX82 can run comfortably without this extension.

The Driver Routines

The driver routines are written in assembler as a single program and provides requisite facilities for TEX82. They are dynamic file binding, file i/o, interactive terminal i/o, and command line access as well as interrupt handling, and retrieving run-time environment information.

* Associate Professor, Hokkaido University Computing Center, Kita 11 Nishi 5, Kitaku, Sapporo, Hokkaido 060, JAPAN

The assembly program uses a set of macros for structured programming. The control flow is represented by macros such as IF, ENDIF, LOOP, ENDLOOP, and so on. Incidentally, the driver program uses looping structure scarcely. There has been a few lines of redundant codes generated by these macros, nevertheless, they helped much in programming the driver routines by clarifying the control structure of assembly codes.

System Dependent Modules

Modules concerned with file name generation are marked "system dependent" in the TEX82 program, and I had to write these parts from scratch as the change file. TEX82 deals with several distinct directories/areas, to which files are placed. There is no counterpart in VOS3, instead, they roughly correspond to system files and user files.

Different convention for generating an external file name from a name parameter given by terminal or input text is employed for each kind of files TEX82 is using. In general, the name syntax is similar to that of external file names. There is subtle difference for some convenience. For example, a name may include lower case letters. A member name is appended to a name parameter separated by a colon, if necessary. A name beginning with '#' represents a system file and is used only by dropping '#'. The rest of names represent user files, to which user identifier is appended in front, and file type is appended to rear, if that name has no file type tag. There is slight difference in the file search order for each kind of files. A file can be bound by an ALLOCATE TSS command or, in batch mode, by a Job Control Language DD statement prior to TEX82 run, and this binding precedes the file search order except for \input, \read, and \write files. A user may control file allocation freely depending on his own interest. In batch mode, terminal i/o can be also bound to arbitrary files, since it is using QSAM interface.

DVIwrite Program

A DVI file output by TEX82 is converted into visual form by the DVIwrite program. This program was developed by rewriting the DVItype. I intended that this program would work system independently as much as possible. The DVIwrite program is consisted of two major parts; the system independent part and system dependent plotting library. The system indepedent part translates a DVI file into vector format, that is, a collection of vectors representing consecutive black dot patterns. A system dependent plotting library draws the vectors as visual form. This part can be substituted for different output devices: laser beam printer, CRT display, facsimile, etc. It seems this strategy is more appropriate for the coming GF font format than the current PXL format.

Several plotting libraries aimed at HITAC CRT terminal, Tektronix graphic terminal, and laser beam printer were developed. Among them, the library for the laser beam printer is the most esoteric. The principal use of the laser beam printer is to print large amount of standard format listings and documents. The interface to the device defined by VOS3 is not appropriate for graphics output such as this. There is a clumsy interface to output graphical information on this device, namely, "forms overlay generation" interface. The interface works as a user own code exit, so a Pascal routine cannot drive this interface directly because of different run-time stack convention. I decided the plot library would work in the way that: the initialiaztion routine attaches a separate task, and this task will drive the interface directly; the rest of plot libraries and the task communicate with each other through synchronization operations WAIT and POST. The interface runs rather slow, since it uses VOS3 sorting utility programs. Consequently, the DVIwrite program is not suitable for interactive processing. The document now you are reading is output through this interface.

Some Statistics

The TEX82 Pascal source code generated amounts to over 5,100 lines and 18,400 statements. The VOS3 Pascal compiler is very slow, it takes 120 second cpu time to compile it. The memory size is over 882 kilobyte including Pascal run-time library, but disregarding run-time stack. The TEX82 run seems very fast, it takes no noticeable time in preparing this manuscript. In fact, it takes about 2 second cpu time. At least one bug was found in VOS3 Pascal compiler while TEX82 was being validated.

DVIwrite is rather slow. It took about one minutes of cpu time to typeset one page of this manuscript. This may be remedied by writing dot-vector-conversion routine in assembler, or by using GF font files. As I obtained 300 dpi font tape and the laser beam printer is 240 dpi, the output is slightly enlarged 25 %.

Finally, the whole job is done by myself in one man-year effort.

Unix TEX Site Report

Richard Furuta

With the beginning of a new academic year, many of us are changing jobs and moving to new parts of the country.

Howard Trickey has completed his Ph.D. at Stanford and is now working for AT&T's research laboratories in Murray Hill, New Jersey. We would be remiss if we failed to acknowledge the great contribution that Howard has made to the Unix TEX world. Howard is responsible for the Unix port of almost every one of the programs in the generic TEX distribution. His influence may also be found in the design of the programs, themselves. His good taste in program design and high standards are clearly reflected throughout the entire Unix TEX distribution. Unfortunately for us, his new position means that he will not be taking as central a role in maintaining the Unix TEX distribution and we will greatly miss his contributions.

By the time this note reaches print, I hope to be comfortably settled into my new position in the Computer Science Department at the University of Maryland, College Park. It is our intention that the Unix TEX distribution will continue to be homed at the University of Washington, at least for the present. As noted in a previous issue of the TUGboat, Pierre MacKay has assumed many of the administrative duties associated with maintaining the distribution. Requests for the tape and queries about the distribution should continue to be addressed to him.

The Unix TEX distribution is very much a collective effort. A few days ago, I decided to put together a mailing list of those people who have contributed directly to this TEX distribution. I was pleasantly surprised to discover that over 30 individuals have contributed material to the tape! We welcome your further contributions.

As the popularity of TEX increases, I believe it important to maintain the open nature of the distribution of material on the various TEX distribution tapes. As examples, we have consciously resisted placing binary-only material onto the Unix TEX tape, providing sources for all material, and we have tried to keep the distribution costs as low as we can. We may have to modify the distribution mechanisms later this year as Pierre's and my work loads increase, but we will try to continue to provide a comprehensive distribution at as close to cost as possible.

In news this time, Mike Urban has provided us with a version of TEX for the Pyramid that relies on a version of the Pyramid Pascal compiler that has not yet been released (as of early September, when this is being written). At press time, we are still in the process of determining how best to merge this material into the Unix TEX tape. If you are interested in this port, please contact Pierre MacKay for further information.

Since the last report in the July TUGboat, Paul Richards of the University of Illinois has continued to supply new versions of the WEB METAFONT for Unix as the development work continues at Stanford. I expect that we will have incorporated the new CM fonts into the distribution by the time this reaches publication. A Unix version of Oren Patashnik's **bibtex** (with change file from Howard Trickey) was added to the tape in mid-June. The **bibtex** program allows reference within LATEX to bibliographic listings that are included in a Scribe-like bibliography file. Also, Rusty Wright of U. C. San Diego has supplied a separate program, named **r2bib**, that translates **refer**-format files into **bibtex**'s format.

In device drivers, Scott Simpson of TRW supplied a number of small fixes to his driver for the QMS 800 and 1200 laser printers. Chris Torek of the University of Maryland continues to enhance his set of device drivers for the Imagen laser printers and for the Versatec. Most notably, the Imagen support now includes the 12/300 in addition to the 8/300. Further, the positioning code in Chris' drivers was updated to correspond to the current **dvitype** definitions, and **dviselect**, which allows extraction of individual pages from a DVI file, was enhanced to allow specification of negative page numbers.

Neal Holtz of Carleton University has provided a driver for the Apple LaserWriter (which uses Adobe's PostScript printing language) and I expect that it will be on the tape before this note reaches press.

Finally, as this note is being prepared, we are in the midst of updating the version of TEX on the tape to be version 1.5. This process should be long completed by the time that the publication reaches you. A new version of LATEX is also expected soon and will be included when it is released.

Mike Harrison of U. C. Berkeley sends along a note, reproduced below, describing the collection of TEX-related software that they have been developing. This software, targeted primarily for the SUN workstation, augments the material on the Unix TEX tape and consequently may be of particular interest to those of you running TEX on the SUN. As obtaining the material requires signing of a license

agreement, Berkeley is distributing the material themselves, so you should contact Professor Harrison for a copy of the license and for information on the distribution charges (the information we have is that these charges are in the neighborhood of $100).

News from Berkeley

The Berkeley VORTEX* project has as its primary goal the design and implementation of a new document processing environment that allows the user to edit/preview TEX documents interactively at both the source level (ASCII representation) and the proof level (bitmap representation), whichever is more convenient. Furthermore, any changes made to one representation will be propagated to the other automatically. There are two basic requirements derived from that goal: (1) VORTEX and TEX must generate identical outputs, and (2) the user should feel more comfortable with VORTEX than with the current disintegrated environment. The design is expected to be complete by the end of this year. prototype will be implemented starting 1986.

The VORTEX group has been developing TEX-related software since early 1984. There is a new distribution of their work now ready for public release. It consists of a tape containing a number of programs that greatly facilitate using TEX and related systems for high quality document preparation. Here is an overview of the major subsystems:

1. `dvitool` is a previewer for `DVI` files that runs on the SUN workstation. This system is very robust, handles arbitrary `DVI` files, and provides a great many features. It is a full tool in the sense of the SUN window system and can be adjusted to any size the user finds appropriate. It is possible to keep a small window on the screen for previewing at the same time a source window is present. This is extremely valuable in debugging. Changing the view you have of a page is instantaneous.
2. `texdvi` is a program that runs TEX and previews the results using `dvitool`. If the tool does not exist it is started, if the tool exists it is opened and the file is read into it automatically. `latexdvi` and `slitexdvi` are similar systems for LATEX and SLITEX respectively. This is actually one program and would work with your own version of "FooTEX" as well (by linking `texdvi` to `foodvi`, for example).
3. `pxtool` is a SUN-based font editor for `PXL` files. It is similar in spirit to `icontool` or `fonttool`. A graphics window is available and an image of the font is shown with the pixels depicted on the screen. Using the mouse, one is able to edit pixels. There is also a "show mode" in which the finished character is displayed on the screen. This tool is very useful for creating and editing fonts.
4. FONTS: A rather complete set of fonts is available for TEX and LATEX in the sizes needed for the previewer (note that SLITEX fonts are not included in this distribution). These are regularly in use at Berkeley and rarely have people run into missing font problems (`dvitool` responds gracefully to missing fonts). These fonts, mostly supplied by the UNIX TEX distribution at the University of Washington, are somewhat bit-tuned using `pxtool` for the SUN screen.
5. `bibtex.ml` is a very large macro package for Gosling Emacs that greatly facilitates the preparation of `.bib` files for document preparation. This is intended for use with LATEX and BIBTEX. The user selects the type of reference intended such as an article and the program provides fields to be filled in, copies fields from previous entries, provides various kinds of checking and assists you in other ways. One particularly useful option is preparing a draft bibliography that includes numerical references, symbolic references and a formatted version of the entries. Another of the options allows previewing on the SUN or printing on any of your local printers. This particular system is not SUN specific although it does interface nicely with `dvitool` mentioned above. Also it is expected to be ported to GNU Emacs in the near future.

If you are interested in the this distribution, please write to:

Professor Michael A. Harrison
Re: VORTEX Distribution
Computer Science Division
571 Evans Hall
University of California
Berkeley, CA 94720

or through the net, to: vortex@berkeley.arpa.

---

* for Visually ORiented TEX.

VAX/VMS Site Report

Barry Smith

This issue's deadline finds us in the midst of too many projects, with none of them complete. On the VAX side, there's a new (1.5) release of TEX, the first release of METAFONT, and the new improved Computer Modern typefaces. Thanks, Don (and crew), for all of these — best wishes for your sabbatical. David (Kellerman) says he's working on an LN03 driver, too.

I've been working on the MacRailroad — Macintosh TEX is getting up steam. Thanks to the Stanford consortium office, we were able to demonstrate Macintosh TEX driving an Apple LaserWriter at the TUG meeting. By the time you read this, we should have an 'integrated' TEX able to switch from input editing to output display and printing (and back) at the click of a mouse.

And, my wife Annie has been busy — we're expecting our first child early next year. (It's a boy!)

Maybe they'll all be ready by Christmas.

"small" TEX implementations

Computer	Processor	Contact	Organization, Address
Hewlett-Packard 1000	16-bit	John Johnson	JDJ Wordware, Box 354, Cupertino, CA 95015; 415-965-3245
Hewlett-Packard 3000	16-bit	Lance Carnes	TeXeT, 163 Linden Lane, Mill Valley, CA 94941; 415-388-8853
DEC PDP-11/44 Plexus, Onyx IBM PC	16-bit[1] Z8000[1] 8086/88[1]	Dick Gauthier	TYX, 11250 Roger Bacon Dr., Suite 16, Reston, VA 22090; 703-471-0233
Apollo	MC68000	Thom Hickey Bill Gropp Pierre Clouthier Jim Sterken	OCLC, Box 7777, Dublin, OH 43017; 614-764-6075 Dept. of Computer Science, Yale University, Box 2158, Yale Station, New Haven, CT 06520; 203-436-3761 COS Information, 5647, rue Ferrier, Montreal, H4C 1V4, P.Q.; 514-738-2191 Textset, Box 7993, Ann Arbor, MI 48107; 313-996-3566
Hewlett-Packard 9836	MC68000	Jim Crumly	Hewlett-Packard, Box 15, Boise, ID 83707; 208-376-6000 x2869
Sun Microsystems	MC68000	Jim Sterken Pierre MacKay	Textset, Box 7993, Ann Arbor, MI 48107; 313-996-3566 University of Washington, Computer Science, FR-35, Seattle, WA 98195; 206-545-2386
Cyb	MC68000	Norman Naugle	Mathematics Dept., Texas A&M University, College Station, TX 77843; 409-845-3104
Apple Macintosh	MC68000	Barry Smith David Kellerman	Kellerman & Smith, 534 SW Third Ave., Portland, OR 97204; 503-222-4234
Masscomp	MC68000	Norman Naugle	Mathematics Dept., Texas A&M University, College Station, TX 77843; 409-845-3104
Synapse	MC68000	Dick Wallenstein	Comcon, 5 Underwood Ct., Delran, NJ 08075; 609-764-1720
PERQ/IQL		Jaap van 't Ooster	Océ, St. Urbanusweg 43, 5900 MA Venlo, Holland
IBM PC, XT, AT	8088, 80286	Lance Carnes	Personal TEX, 20 Sunnyside, Suite H, Mill Valley, CA 94941; 415-388-8853
IBM XT, AT	8088, 80286		Addison-Wesley Publishing Co., Educational and Professional Technologies Division, Reading, MA 01867; 617-944-6795
IBM XT, AT	8088, 80286[2]	Ronny Bar-Gadda	446 College Ave., Palo Alto, CA 94306; 415-326-1275

[1] not TEX82 [2] in progress or recently completed

Typesetting on Personal Computers

Assembling a Moderately-Priced, High-Performance Clone of the IBM PC for Running TEX

M. Pfeffer and A. Hoenig

This new column is dedicated to bringing to users of all brands of personal computers the latest information on products related directly to TEX (including: new implementations, printer drivers, and screen previewers), as well as singling out those products from the general computer marketplace that are of special interest to people who set type using personal computers. (Let's hear what you've learned from your experiences, and what you'd like to see in future columns.)

The focus of this issue's column is on building a moderately-priced, high-performance IBM PC clone from sub-assemblies. The information should prove useful not only to users building the complete system, but also those users needing to expand their present systems to handle TEX. This $2,000 computer has a 20 megabyte hard disk, 640 k of RAM, and runs TEX in 30% less time than a PC. Assembly takes just a few hours, needing only a set of screwdrivers (but a set of small nutdrivers helps). Here's the list of the components, followed by the list of suppliers and the assembly instructions. (Footnotes are references to reviews.)

Hard Disk: Although you could work with a 10 Mb drive (TEX needs 5 Mb), it pays to get a 20 Mb drive: the difference in price is small.

PC's Limited sells 20 Mb drives for $495, with a one-year warranty. They ship from a variety of manufacturers; the drive I received was a half-height Seagate, with a Western Digital controller. I timed it as having a 75 ms average access time.[1] [2]

If you are adding a hard disk to your present PC, you should replace the PC's 65.5 Watt power supply with a higher-capacity model (see below).

(If you plan to run an operating system other than DOS, you'll probably need to spend an additional $75 for a Xebec controller.)

Operation Hints: Put '`prompt $p$g`' in your `autoexec.bat` file; put '`buffers=16`' in your `config.sys` file; to avoid accidentally formatting the hard disk, rename `format.com` to `formatxx.com` and create a file named `formatA.bat` containing the line '`formatxx A:`'—to then format a floppy disk, you use '`formatA`'; park the disk's head before moving the computer (and also, preferably, before turning off the computer); review the '`/p`' and '`/w`' options to the `dir` command; routinely backup your files (you can download the backup program `bac.com`, described in *PC Magazine*, vol. 4, no. 17, pages 197–205, August 20, 1985, by calling [212] 696-0360, where you will also find the `diskp.exe` program I used to time the drive). Also, look into some of the handy file utilities that make managing a hard disk easy.[3]

Motherboard: For $670, ACS sells a tested motherboard[4] which runs at 8 MHz (a PC runs at 4.77 MHz), with 640 k of RAM, built-in floppy-disk controller, a battery-driven clock/calendar, two serial ports, one parallel port, and six expansion slots (spaced as on an XT). The board comes with a one-year warranty; the current version of its ROM is 2.3, and the current version of the included diskette (containing a RAM-disk program, and the program to set the clock/calendar) has 'version 2.2' on its label.

You'll need to order a floppy-disk cable from ACS (for $20) to connect the board to the floppy-disk drive, as well as a speaker ($5, with cable). I also bought a serial cable from ACS ($15; it comes with a DB-25 connector on its end) to allow the use of an external modem.

(A one-megabyte version of the board is available for $730; this would allow you to use 192 k of the memory above DOS's 640 k limit as part of a RAM-disk, using the included RAM-disk program (the remaining RAM is preempted by address conflicts). Placing a copy of your editor and manuscript in RAM-disk would allow fast corrections to the manuscript between runs of TEX.)

Drawbacks: The system lacks power-on diagnostics; the manual is sparse, and no technical manual is available; changing between 8 MHz and 4.77 MHz requires opening the case to move a jumper.

If you plan to write programs in Basic, you'll need to purchase a stand-alone Basic, as no Basic kernel is present in ROM.

---

[1] "Ten by Ten," *PC Tech Journal*, vol. 2, no. 5, pages 53–63, November, 1984.

[2] "Fixed-Disk Benchmarks," *ibid.*, pages 64–70.

[3] "The Enhanced AT: A Personal Vision," *PC Magazine*, vol. 4, no. 13, pages 132–133, June 25, 1985.

[4] "Personal Computers: Kit Computer Turns Out To Be a Sane Idea," *The New York Times*, Section C, May 28, 1985.

Compatibility: In the 8 MHz mode, I tested: both implementations of TEX; PC-Write (my current editor); The Word Plus (a spelling checker); PC-Talk; Turbo Pascal 3.0; Masm; and the Norton Utilities. I was able to bring up (but didn't test extensively): FinalWord's editor; Edix; Microsoft Word; Epsilon; WordStar; and Trace86.

Most incompatibilities surface in the 8 MHz mode. Programs that depend on software timing loops will fail. Some cards can't keep up with the 8 MHz speed. One user of the ACS board told me that his Hayes *internal* modem wouldn't work (Hayes told him what needed to be changed to fix the problem), and that he had to patch Crosstalk for the program to run at 8 MHz.

(If you purchase the Xebec 1220 hard-disk controller, which incorporates a floppy-disk controller, you'll need to disable the floppy-disk controller built into the ACS board—see the note in the ACS manual, under section 6.12, explaining which chip to replace; the replacement chip costs $15.)

Operation Hints: With a Hercules card in the system, use `clklpt1` to set the clock initially, and to transfer the correct time to DOS—the ACS manual incorrectly implies `clklpt2` should be used.

Some programs can't cope with 640 k of RAM, and give an 'out of memory' message, even though they have plenty of room. One solution is to create a RAM-disk, to reduce the amount of memory available to the program.

Floppy-Disk Drive: I chose the CDC half-height drive, based on a review[5] and my favorable experience with the CDC full-height drives. (But CDC is dropping out of the 5 1/4-inch floppy-disk drive market.)

Power Supply: I purchased a 150 Watt supply from PC's Limited for $119. I've since noticed an ad for a 200 Watt supply from CC&C.

Case: Purchased from G & L, the case was satisfactory, though some of the screws were improperly machined. The case won't allow the plug from an IBM keyboard to reach the motherboard's jack, because of the grip ring molded onto the plug; if you want to use the IBM keyboard, you'll need to use a keyboard-extension cable (judging from a picture, CC&C's MSC-I case would accommodate the IBM keyboard's plug).

Next time, I'll try a case with a flip-up top, like the one sold by JDR Microdevices for $59.95.

Display Adapter: I wanted a card that included Hercules monochrome-graphics emulation, for use with the new TEX screen previewers. I ordered a card made by Mitsuba from Computer Systems Planning, at $125. It ran Microsoft's Word in Hercules mode, and Hercules HBasic (on an IBM PC).

Display: Studies indicate better user-performance with yellow displays, with one study revealing the worst performance with an orange display;[6] yet manufacturers use the term "amber" to cover pumpkin-orange through canary-yellow. Restricting the search to yellow displays still leaves many displays from which to choose,[7] and other criteria to apply.[8] (I've always been satisfied with the quality of IBM's green display, but in a side-by-side comparison, a yellow display does seem less harsh.)

The display I tested, the Taxan 122, uses a yellow, long-persistence (PUL) phosphor, to minimize flicker, but the trade-off is a more noticeable afterimage when scrolling.

(Flicker is more discernible when a display is viewed out of the corner of your eye. To check the flicker of a display, as well as imperfections in the phosphor coating, configure a program to work in reverse video; for PC-Write, a freely-copyable editor which can be set to work in reverse video, see Appendix A of its manual.)

Drawbacks: As delivered, the top two scan lines would overlap when the top line was in reverse video, giving the appearance of a single, bright line; adjusting the vertical-line control corrected this. (I also tweaked the other controls to improve the image.)

Operation Hints: For a sharper image, I set the brightness to its near-minimum position, and used the contrast control to adjust the intensity.

Anti-Glare Screen: I purchased a mesh screen, called SuperScreen, from R+R Direct, to reduce glare off the display. The screen imparts a jet-black background to the characters; this makes the characters stand out. I also tested a polarized panel made of plastic, but found it produced a worse glare from the panel than I had off the display.

Drawbacks: The mesh causes a slight degradation of the image, and superimposes its own dot pattern on the characters.

---

[5] "Double Your Driving Pleasure," *PC Magazine*, vol. 4, no. 10, pages 133–137, May 14, 1985.

[6] "Maximize Your Computing Comfort and Efficiency," *Computers & Electronics*, vol. 21, no. 4, pages 35–48, April, 1983.

[7] "Buyer's Guide to Monitors," *Computers & Electronics*, vol. 22, no. 12, pages 56–59, December, 1984.

[8] "Video Signals and Monitor Design," *ibid.*, page 53.

Keyboard: The low-cost keyboard I tested differs from the IBM keyboard: it has a soft touch; the keys do not click; it has separate numeric and cursor keypads, horizontally-arranged function keys, and a standard letter-layout (TEX users may be the only people who prefer IBM's non-standard placement of the backslash and open quote on the PC's keyboard. In the AT, IBM uses the standard office layout).

I purchased the keyboard from ACS for $125 (JDR sells what appears to be the same keyboard for $100). Minor variations on this keyboard exist: some, like the one I received, have the caps-lock key next to the 'a' key, while others have the control key next to the 'a' key; some have the PC's placement of the backslash and open quote—be sure to specify your preference when ordering.

Disadvantages: Though some users claim that the soft touch increases their typing speed, I find the cheap feel of the keyboard to be too mushy. The legends on the keys are only surface markings, and will rub off with use.

Surge Suppressors: I know of four instances where computers were fried during lightning storms. Protect your equipment with surge suppressors—one on the power line, and another on the phone line to the modem.

A Curtis Ruby (PC Connection, $59), protects my power line. A single switch controls its six outlets.

The new Radio Shack catalog lists a phone-line protector (43-102), for $12.95. Because phone-line protectors often protect only the red and green wires, I disconnected the other two wires (black and yellow) from the jack that feeds into the protector. (The black and yellow pair is unused in most homes with single-line service; the pair is used to activate the line-in-use light in some phone installations, or carries the second line in a two-line system.)

(Incidentally, owners of Compaq and IBM systems can exchanged fried motherboards for replacement boards, at less than one-third the cost of a new board.)

Cables: I ordered Curtis printer and modem cables from PC Connection, at $19 each. The Curtis cables are completely shielded (including the plugs), to prevent stray signals from getting in or out of a cable.

Accessories: Some odds and ends to make your system more comfortable: tilt/swivel base for the display; copy stand (without any magnets); ergonomic chair; flip-top floppy-disk file (such as the one made by MicroComputer Accessories); glare-free lighting; and dust covers.

Software: Aside from TEX and an editor, you'll need a copy of DOS.

Setup: To minimize strain, the keycaps of the home row of your keyboard should be about 27 inches from the floor: the goal is for your wrists to be level with, or slightly below, your elbows—the standard 29½-inch table height is too high. (My keyboard rests on a small folding coffee-table, in front of the desk that supports the display.)

The display should be leveled horizontally, and propped up to bring its top edge to just below eye level. This also gives you room to place the copy stand directly below the display, minimizing head-turning as you look back and forth.

Suppliers

When placing your orders, ask the expected ship date, and the difference in cost between ground and air shipping.

PC's Limited, in TX: [800] 426-5150 (from Texas: [512] 452-0323). 20 megabyte internal hard disk for a PC (you can specify a Seagate), $495; 150 Watt power supply, $119. (Free ground shipping; about $14 by second-day air.)

ACS, in TX: (Advanced Computer Solutions) [214] 247-5151. Motherboard with 640k, $670; speaker, with cable, $5; floppy-disk cable, $20; serial cable, $15; keyboard, $125. Address orders to Tom Langley. (About $17 for second-day air.)

PC Connection, in NH: [800] 243-8088 (outside continental US: [603] 446-3383). Half-height CDC floppy-disk drive, $89; Curtis Ruby surge suppressor, $59; [Curtis printer cable, $19; Curtis modem cable, $19]. (Second-day delivery: $2.)

G & L, in CA: [800] 523-8750 (from CA: [714] 758-8600). XT case, $49. (Second-day delivery: $24; ground: up to $13).

Computer Systems Planning, in NY: [212] 213-9125. Mitsuba display adapter, $125. This company also sells assembled systems, based on the ACS board; contact Paul Wolotsky.

LogicSoft, in NY: [800] 645-3491 (from NY: [516] 249-8440). Taxan 122 amber display, $119 (shipping by ground: $3). LogicSoft will beat advertised prices by $10; I used this policy, having them beat the $129 price advertised by Silicon Specialties ([800] 354-7330).

R+R Direct, in OH: [800] 654-PLUS (from OH: [800] 545-PLUS). SuperScreen (A5193 for Taxan 122), $29.95. (Not a discount house.) Excellent customer service.

Other Companies: Your Personal Computer Store, in TX, [409] 740-3223. Xidex floppy-disks, $12 for a box of ten, double-sided, double-density disks. Free ground shipping for orders over $50; $4 for second-day shipping of three boxes.

JDR Microdevices, [800] 538-5000 (from CA: [800] 662-6279).

CC&C, in CA, [818] 576-1621.

MicroComputer Accessories, in CA, [213] 641-1800.

Quill Corporation, in IL, [312] 634-4800. Computer and office supplies. Good price if item is on sale (a frequent occurrence). Excellent customer service.

Assembly

In the directions that follow, my use of "left" and "back" assumes that the case is oriented as if in operation on a desk in front of you, and that the motherboard is oriented as it would be when inside the box: component-side up, card-slots in the back-left corner.

Prepare the Case: Put aside six of the chrome screws (the ones with hex/Phillips heads)—they'll be used later to attach the top cover to the black base, and to secure the two cards. Remove the back card-support cage (the one with eight cutouts). Attach a chrome blanking plate over each cutout, except over the first cutout (the left-most cutout when held as if already mounted inside the case—screw-holes up), and except over the sixth cutout (I used the remaining chrome hex-head screws for this, and when I ran out, finished with the black screws with the built-in washers). Loosely attach the cage inside the box (allowing it to slide), using four black screws (the ones with the built-in washers); the cage will be tightened later.

Attach a black plastic edge-guide (which stabilizes the front edge of a card) to the left-most position inside the front of the base; insert one of the tiny black screws from the inside of the case, through the top hole in the guide, and secure with a hex-nut on the outside of the case. (The edge-guides that came with my case overlap more of a card than the snap-in guides that normally come with cards, but this didn't create a problem with the Mitsuba card.)

Drop the speaker into its position at the front of the base. Bring its cable around the lower-left edge of the base's front panel, into the inside of the base; if the cable isn't long enough to reach the near left drive shelf (close to the point where the speaker's connector is plugged onto the motherboard), you'll have to route the cable through one of the grill cutouts behind the speaker, and force the speaker down into place. I used a dab of epoxy to secure the speaker.

Attach the small, black, oval blanking plate inside the rear of the box, to cover the left of the two D-connector cutouts. Install the D-connector from the serial cable in the right cutout, and route the other end of the cable out the large circular hole, to keep it out of the way for the moment.

Prepare for the Motherboard: The board is held in place by nine brass stand-offs, hex-nuts, and black screws. To eliminate improperly-machined hardware at this point, thread the hex-nuts onto the stand-offs, and insert the tiny black screws into the other end (if you don't have enough tiny black screws, use the black screws with built-in washers).

Turn to Fig. 1 of the ACS manual, and make the following annotations (it's easier to find the proper connection points by working from this drawing, rather than the legends printed on the circuit board, as it's sometimes difficult to decide if a legend on the board refers to the connector to the left of the legend or the connector to the right of the legend): E1 is the speed jumper; P1 is the connector for the two power leads from the power supply; S1 is the set of DIP switches; P3 is the connector for the floppy-disk drive's cable; P6 is the connector for the serial port's cable (`COM1:`); and P2 is the connector for the speaker.

Configuring the Motherboard: To discharge any static you may be carrying, touch a large metal object; then remove the motherboard from its anti-static bag, and rest the board on top of the bag.

To set the board to its 8 MHz speed, move the jumper on the E1 block so it connects pin 2 to pin 3.

Refer to section 7 of the manual to set the DIP switches; on my board, I set 5 and 6 'off' (for monochrome display), and I set 7 and 8 'on' (for a single floppy-disk drive). Verify that switches 3, 4, and 9 are properly set for the amount of memory on your board (see section 7.03).

Installing the Motherboard: The threaded ends of the brass stand-offs are inserted from the bottom of the board, and secured with hex-nuts on the top (component side) of the board, but to prevent the mounting hardware from short-circuiting the metallic traces that run near some of the board's mounting holes, you must use the insulating red washers. Ideally, insulating washers would be placed above and below every hole, but if your case doesn't come with enough washers to do this, use what washers you have in these ten essential locations (orient the board component-side up, with

the card slots in the back-left corner; a row runs left-to-right, a column runs back-to-front): back row, middle column: top side; middle row, all three columns: top and bottom; front row, left corner: top and bottom; front row, middle column: top. You can use a nut driver to hold the hex-nuts as you tighten the stand-offs, but don't try to hold the nuts with a pliers—pliers are too likely to slip off the nut, and damage a trace.

After installing the stand-offs, hold the board above the case's back-left corner, with the middle column of stand-offs just inside the left lip of the case, and the back of the board just touching the back of the case; lower the board into the case (the left-most column of stand-offs will be resting outside the case). Slide the board to the right, until the left-most column of stand-offs touch the outside of the left lip of the case. Pull up on left drive shelf; while pulling it up, lift the left edge of the motherboard up, until the left-most column of stand-offs are above the left lip of the case; slide the motherboard right, until the stand-offs are to the right of the case's lip, then lower the board onto to bottom of the case. While maintaining your pull on the drive shelf, slide the motherboard into position. Release the drive shelf.

To temporarily counterweight the base, place a heavy book in the right drive shelf (volume 2 of *The Art of Computer Programming* does nicely). To install the screws into the left column of stand-offs, slide the base so that the left side of the case extends over the edge of the table. Install the screw in the middle row first. Don't tighten any of the nine screws until all are in position. After installing the three screws in the left-most column, move the base further off the table, until the middle column of holes is accessible (take care that the case doesn't fall off the table) and install the screws in this column, again starting with the middle screw. Move the base back on the table. Remove the book. Connect the free end of the serial cable to P6 (as always, the side of the cable with the red stripe goes near pin 1 of the motherboard's connector—be sure to position connectors squarely on their double row of pins). To insert the screws in the right-most column, turn the base over, or stand it up on its back panel. Tighten the nine screws.

Clean those areas where you intend to place the self-adhesive feet, and then attach the feet.

Connect the plug from the speaker to P2; route the cable down and out of the way. Connect the floppy-disk drive cable to P3 (red stripe on cable to pin-1 side of connector), and allow the cable to run off to the side.

Installing the Power Supply: Of the six plugs coming out of the power supply, four are identically shaped. Of these four identically-shaped plugs, two will be used to power the two drives; the other two remain unused. The remaining two (larger) plugs will be connected to the motherboard at P1.

Remove the four round-head screws from the back of the power supply—these will secure the supply in place. Slide the power supply in from the right. When it's half way in, connect the two larger plugs to P1 (these plugs are keyed with small nibs, to prevent you from incorrectly switching them around). Slide the supply in the rest of way, then slide it forward toward the drive shelves, and then back against the rear of the case (this causes the lips in the bottom of the supply to engage the lips in the case). Use the four screws to secure the power supply.

Installing the Hard Disk: (Read the directions that come with the drive.) Partially slide the drive into the right drive shelf. Attach the two ribbon cables from the controller, which is still outside the case, to the drive (the side of a cable with the red stripe goes closest to the notched side of an edge-card connector on the drive). Attach the power lead from the power supply (this D-shaped plug will fit only one way). Slide the drive in the rest of the way. Insert the screws loosely (I used some spare screws to secure the drive on both sides). While pushing the drive back against the front of the case, tighten the screws.

Installing the Floppy-Disk Drive: (Read the directions that come with the drive; the drive comes configured as an 'A' drive, which is what we want.) Partially slide the drive into the left drive shelf. Attach the connector from the power supply. Attach the connector that's at the end of the cable coming from P6 to the drive (the side of the cable with the red stripe, and twisted segment, goes to the notched side of the drive's edge-card connector). While gently holding the cable from P6 down against the motherboard, slide the drive in the rest of the way (the drive sits atop P6's connector and ribbon cable). Loosely insert the two screws (by hand) into the left side of the drive (if you drop a screw, take care not damage the board when retrieving the screw). You're now about to tighten the screws—but to avoid the danger of dropping your screwdriver onto the motherboard (and damaging the delicate metallic traces), tie a loop of string loosely around your wrist, and tie the other end around the screwdriver; this way, if you accidentally let go of the screwdriver, it won't fall onto the

board. While pushing the drive back against the front of the case, tighten the screws.

Inserting the Display Card: Slide the Mitsuba display adapter into slot 1 (the left-most slot). Adjust the card cage so the connectors on the card are centered in the cutout. Use a hex-head chrome screw and the screwdriver with the wrist strap to attach the mounting bracket of the card to the cage. Tighten the cage to the base.

Insert the Hard-Disk Controller: Slide the controller into slot 6. Using the screwdriver with the wrist strap, secure the card's mounting bracket to the cage, with a hex-head screw.

Completing the Assembly: Insert a black plastic blanking plate above each drive.

Neatly fold the cables together, and slide them into the space between the drives and the power supply. (I used 8-inch nylon wire ties, Radio Shack 278-1642, to keep the wires together.) Push the cables from the hard-disk controller down and against the controller, so they won't snag on the lip that's beneath the center of the top cover, as you slide on the cover.

Slide on and secure the top cover; check that the cover's lip has engaged the lip on the top of the base. Connect the display and keyboard.

To attach the anti-glare mesh, place the Velcro pads on the top and bottom (horizontal) parts of the frame—the vertical sides of the frame don't sufficiently overlap the display's case to give a firm grip.

Follow the directions that came with the hard disk on partitioning and formatting the drive (my disk had already been partitioned).

Date submitted: October 4, 1985.

Review: The Boston Computer Society's IBM PC & Compatibles Technical Word Processor Review

Barbara Beeton

The September 11 meeting of the PC Technical Group of the Boston Computer Society (BCS) was devoted to scientific word processing programs which run on IBM PC and PC compatibles. A ten-month study of such programs preceded the meeting, and culminated in a lengthy, comprehensive report. The authors of the report are Avram Tetewsky, Charles Stark Draper Lab, and Jack Pearson, Avco Systems. TEX, in the MicroTEX and PC TEX implementations, was one of the programs examined in the study.

The aims of the study were: "firstly, to define the needs of the review committee members (which varied across several technical disciplines) as well as those of a wider cross-section of users; secondly, to make these needs known to software vendors — so that they can better understand and respond to user needs; and finally, the committee members themselves wanted to see what was available."

These areas in particular were examined:

- user interface — WYSIWYG vs. markup;
- customizability — fixed or open selection of fonts, macro command language, screen and printer drivers;
- ability to interchange text data with other systems;
- quality and speed of hardcopy output;
- competence in handling technical material, as demonstrated by the ability to cope with a selection of benchmark examples;
- ease of use — how much *brainware* is required.

The results of the analyses appear in short reviews of each program, and in six summary tables. Vendor names and addresses are listed, as are references for further reading. In all, 38 vendors are listed, supporting 36 programs. 10 programs are reviewed in more or less detail (TEX accounts for two of the ten), and summary information is given for 22 others, for which there was not enough time to complete the benchmarks or for which testable copies were not available. Tabular information includes such things as price, hardware requirements, file formats, physical features which can be customized (fonts, printer, etc.), benchmark results, program features available (footnotes, tables, etc.), "add-ons" available (hyphenation, spelling checker, spreadsheet, etc.), and documentation quality. Finally, there is some information on programs available for non-IBM PCs, including CP/M, Macintosh and 68000 systems (there is speculation as to whether TEX can be brought up on the Mac; the latest information available to TUGboat on this and other items has been provided to the authors of the report).

TEX was reviewed by A. G. W. Cameron, Harvard College Observatory (a TEX user with a year's experience), and Jack Pearson (a new user). The review gives a general introduction to TEX, and explains why TEX is not a "word processor" in the same sense as the other programs tested. It examines the technical differences between MicroTEX and PC TEX, and finds them both to be "excellent implementations of TEX". Differences between the two are attributed to the release level (MicroTEX = 1.4 and PC TEX = 1.0) and to the language of implementation (PC TEX is implemented in Pascal and assembly language, and MicroTEX in a C translation). One obscure bug was found in MicroTEX, and both MicroTEX and PC TEX had memory problems when asked to produce Benchmark 10, a composite of the first nine. (The developers were informed of all problems; both companies are now preparing TEX 1.5 for release.) The enthusiasm of the reviewers is evident in abundance, although they admit that considerable "brainware" is needed. The conclusion: "...I am convinced that TEX systems are in a class by themselves. The TEX user has power and flexibility unmatched by any conventional word processor. ... I suspect that soon a lot of people will be using TEX systems for [many kinds of nontechnical documents]. Why? Simply because if one person or business uses TEX, those who don't will look relatively crude in comparison. And no one wants his competition to have that kind of an edge."

The demonstration version of the benchmarks was prepared in TEX by Dr. Cameron, who has kindly provided to TUGboat his source file, which was used to generate the output below. (His article describing the code used to generate Benchmark 7 appears on page 155.)

By arrangement with the authors and the BCS, the report will be published in the *Notices of the American Mathematical Society* early in 1986. (Some other articles on subjects related to TEX and technical word processing will also appear in that publication.)

This review is based on a draft of the BCS report, and some changes can be expected in the published version. Copies of the BCS report may be obtained by sending a check for $8 (payable to him) to

Carl A. Hein
Dunster House, Apartment 7
Swanson Road
Boxborough, MA 01719

Benchmark 1:

Tsang, L., and Kong, J.A., *Journal of Applied Physics*, **51**(7), July 1980, page 3471, equation 110.

$$W^{3\beta}_{m_1n_1n_2}(p_1,p_2) = U^{3\beta}_{m_1n_1}(p_1,p_2) + \int_0^\infty \frac{dp_3p_3^2}{8\pi^3} \sum_n \sum_m \sum_{\alpha_2} \sum_{\beta_2} \sum_{n'} \sum_{n''} (-1)^m$$

$$\times \left(\frac{U^{33}_{m_1n_1}(p_1,p_2)}{p_3^2 - k^2} \right) z_{3m_1n_1} h_n(p_3,p_2) \cdot a^{\alpha_2 3}_{mn(m_1-m)n'n_2} a^{\beta_2\beta}_{-mn(-m_1+m)n''n_2} W^{\alpha_2\beta_2}_{(m_1-m)n'n''}(p_3,p_2). \qquad (110)$$

Benchmark 2:

Papoulis, Athanasios, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1984, page 17.

Unions and intersections The *sum* or *union* of two sets $\mathcal{A}$ and $\mathcal{B}$ is a set whose elements are all elements of $\mathcal{A}$ or of $\mathcal{B}$ or of both (Fig. 2-3). This set will be written in the form

$$\mathcal{A} + \mathcal{B} \qquad \text{or} \qquad \mathcal{A} \cup \mathcal{B}$$

The above operation is commutative and associative:

$$\mathcal{A} + \mathcal{B} = \mathcal{B} + \mathcal{A} \qquad (\mathcal{A} + \mathcal{B}) + \mathcal{C} = \mathcal{A} + (\mathcal{B} + \mathcal{C})$$

We note that, if $\mathcal{B} \subset \mathcal{A}$, then $\mathcal{A} + \mathcal{B} = \mathcal{A}$. From this it follows that

$$\mathcal{A} + \mathcal{A} = \mathcal{A} \qquad \mathcal{A} + \{\emptyset\} = \mathcal{A} \qquad \mathcal{I} + \mathcal{A} = \mathcal{I}$$

The *product* or *intersection* of two sets $\mathcal{A}$ and $\mathcal{B}$ is a set consisting of all elements that are common to the sets $\mathcal{A}$ and $\mathcal{B}$ (Fig. 2-3). This set is written in the form

$$\mathcal{AB} \qquad \text{or} \qquad \mathcal{A} \cap \mathcal{B}$$

Benchmark 3:

Feynman, Richard P., *The Feynman Lectures in Physics*, Addison-Wesley Publishing Co., Vol. 3, page 20-12, Table 20-1.

Table 20-1

Physical Quantity	Operator	Coordinate Form
Energy	$\hat{H}$	$\hat{\mathcal{H}} = -\frac{\hbar^2}{2m}\nabla^2 + V(r)$
Position	$\hat{x}$	x
	$\hat{y}$	y
	$\hat{z}$	z
Momentum	$\hat{p}_x$	$\hat{\mathcal{P}}_x = \frac{\hbar}{i}\frac{\partial}{\partial x}$
	$\hat{p}_y$	$\hat{\mathcal{P}}_y = \frac{\hbar}{i}\frac{\partial}{\partial y}$
	$\hat{p}_z$	$\hat{\mathcal{P}}_z = \frac{\hbar}{i}\frac{\partial}{\partial z}$

In this list, we have introduced the symbol $\mathcal{P}_x$ for the algebraic operator $(\hbar/i)\partial/\partial x$:

$$\hat{\mathcal{P}}_x = \frac{\hbar}{i}\frac{\partial}{\partial x}. \qquad (20.60)$$

Benchmark 4:

Brogan, W.L., *Modern Control Theory*, QPI Quantum Press Inc., Prentice Hall, 1982, page 180, Figure 9.11.

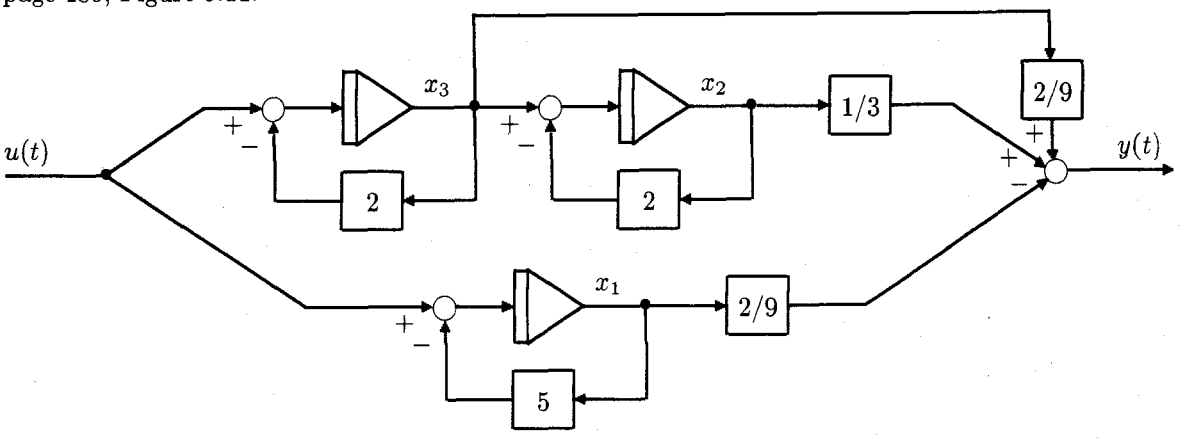


Fig. 9.11

From Fig. 9.11,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -5 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u \qquad \text{and} \qquad y = [-2/9 \quad 1/3 \quad 2/9]\, x$$

Benchmark 5:

Marsden, J.E., *Elementary Classical Analysis*, W. H. Freeman and Co., 1974, page 234, proof of Theorem 2.

Proof: Define the function $G : \mathbf{A} \subset \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}^n \times \mathbf{R}^m$ by $G(x, y = (x, F(x, y))$. Since F is of class C^p and the identity mapping is of class C^∞, it follows that G is of class C^p. The matrix of partial derivatives of G (Jacobian matrix) is

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & & & & & \\ \vdots & & \ddots & \vdots & \vdots & & \vdots \\ 0 & & \cdots & 1 & 0 & \cdots & 0 \\ \dfrac{\partial F_1}{\partial x_1} & & \cdots & \dfrac{\partial F_1}{\partial x_n} & \dfrac{\partial F_1}{\partial y_1} & \cdots & \dfrac{\partial F_1}{\partial y_m} \\ \vdots & & & \vdots & \vdots & & \vdots \\ \dfrac{\partial F_m}{\partial x_1} & & \cdots & \dfrac{\partial F_m}{\partial x_n} & \dfrac{\partial F_m}{\partial y_1} & \cdots & \dfrac{\partial F_m}{\partial y_m} \end{pmatrix}$$

Benchmark 6:

Henry, Allen F., *Nuclear Reactor Analysis*, MIT Press, Cambridge, Mass, 1982, page 495, equation 11.4.19, subequations 4 and 5.

$$iB_r[\tilde{a}_{kl}^n] \equiv \frac{1}{2}(h_{n-1} + h_n) \int_0^R 2\pi r\, dr [\rho_k^{n*}(r)] \frac{d}{dr}[\Psi_l^n(r)],$$

$$[D^n_{r,kl}]^{-1} \equiv \int_0^R 2\pi r\, dr \int_{z_n-\frac{1}{2}h_n-1}^{z_n+\frac{1}{2}h_n} [\rho^{n*}_k(r)][D^{-1}(r,z)][\rho^n_l(r)],$$

Benchmark 7:

Guendelman and Radulovic, "Infrared Divergence in Three-Dimensional Gauge Theories", American Physical Society, **30**, No 6, 15 Sept 1984, page 1347, Figure 13.

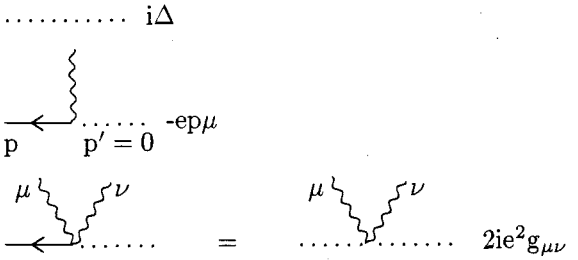
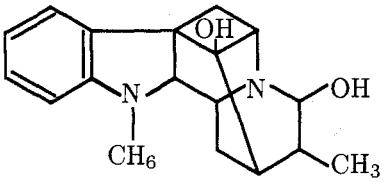


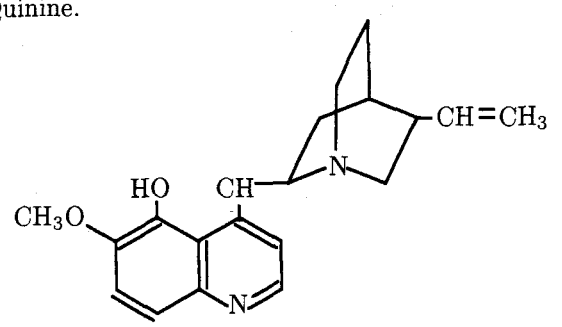
FIG. 13. Feynman rules for the ϕ_c scalars.

Benchmark 8:

Hendrickson, Cram and Hammond, *Organic Chemistry*, McGraw-Hill, 1970, page 1078, Figure 27-6, Ajmaline and Quinine.



Ajmaline



Quinine

Benchmark 9:

The following expressions.

$f_{\underline{Z}}(\underline{Z})$ $\quad$ $f_{\underline{y}}(\underline{y}) \cdot e^{\alpha^\beta}$

Benchmark 10:

Place benchmark examples 1 through 9 in one file. See if:

(1) pagination works, and

(2) the system has enough memory or stack to do the work.

Warnings

The PLAIN Truth: \buildrel

Barbara Beeton

Changes to PLAIN.TeX, though infrequent, are usually somewhat inscrutable. This column will attempt to illustrate with (relatively) simple examples the reasons for some of the changes that have occurred since TEX was "frozen" at version 1.0.

\buildrel

All changes to \buildrel have taken place where the * appears in this "generic" definition:

```
\def\buildrel#1\over#2{\mathrel
        {\mathop{*#2}\limits^{#1}}}
```

A handy little tester for this macro is built right into PLAIN:

```
\def\doteq{\buildrel\textstyle.\over=}
```

Let's use this, plus a simple variation with x replacing =, to show the change history.

(1) The original definition contained nothing but the argument within the \mathop.

$$a \doteq b \dot{x} c$$

This works well enough for \doteq, but it's just luck—single characters within \mathop have their baselines altered so that they align vertically with the axis (rule 13, *The TEXbook* Appendix G, pp. 443-444).

(2) The next iteration inserted \null.

$$a \doteq b \dot{x} c$$

\null is type Ord, which generates space when adjacent to a symbol of type Rel.

(3) \hskipOpt eliminates this space:

$$a \doteq b \dot{x} c$$

however, it requires 3 words of memory, to accommodate stretch and shrink, which aren't needed here.

(4) \kernOpt accomplishes the same thing, using only 2 words of memory.

$$a \doteq b \dot{x} c$$

This is presumably the "final" word on the subject.

(*To be continued*)

Macros

Announcement of LATEX Version 2.09

Leslie Lamport

The end is in sight. LATEX Version 2.09 is now available. This is the final, last, terminal, ultimate version. No more new features. Bugs will be fixed as usual. Some time soon, after people have had a chance to find bugs, this version will be renumbered to be Version 3.0, which sounds much more final than 2.09.

The LATEX manual is now in production, and Addison-Wesley should have it in your local bookstore around the middle of October. Meanwhile, LERRATA.TEX will tell you what has changed. There's not too much that's new in Version 2.09. However, there are a lot of changes in the procedure for obtaining and setting up new versions, so read on.

The primary source of LATEX files is now the <TEX.LATEX> directory on SU-SCORE. My new Arpanet address is lamport@decwrl. (I assume that people will be able to figure out what to append to this address when the net is fully fragmented.)

Users who read the new manual will have some new expectations. Installers will have to see that they are met, or they'll have some unhappy customers. The file LATEX.INS describes what must be done, but here's a short description.

The primary innovation is a *Local Guide*—a document that gives the site-specific information they need for using LATEX. The file LOCAL.TEX contains the LATEX input to produce the document in use at DEC's System Research Center in Palo Alto. The installer must modify this document for their site and arrange for its distribution. (It would be really neat if university book stores could provide the *Local Guide* along with the LATEX manual.) Since LOCAL.TEX is written for an Ultrix system, it will be fairly easy to modify it for an Ultrix or Unix site.

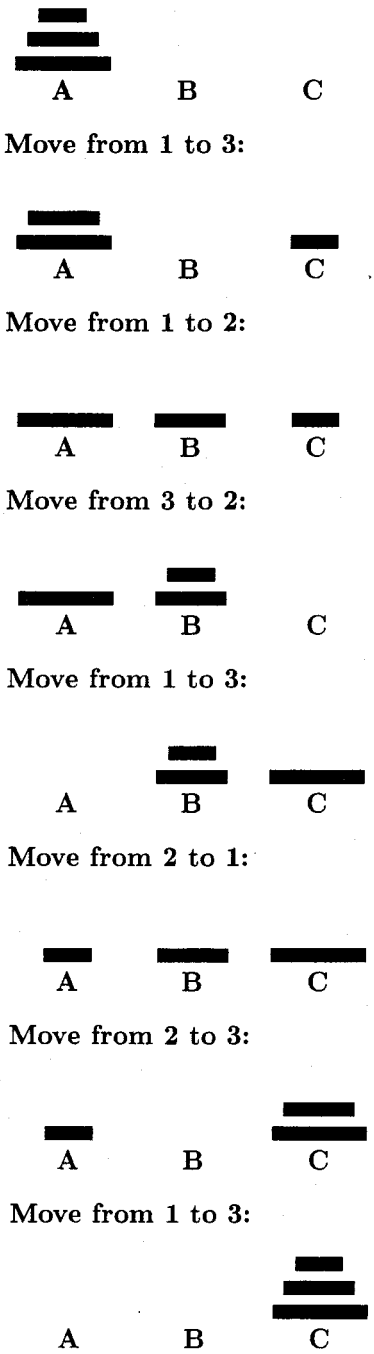
There are also two short files that provide quick introductions to LATEX. The shorter, SMALL.TEX, covers only the most elementary things. It contains a pointer to on-line documentation of how to use LATEX at the specific site. The installer must modify that pointer and create the pointed-at documentation. The file SAMPLE.TEX is a more complete tutorial.

Another goodie that's in this release is the `proc` document-style option. It produces double-column conference proceedings format on $8\frac{1}{2} \times 11$ paper. (Instead of sending in your camera-ready copy on those large sheets that they reduce by 25%, you can produce it on a high-quality output device and send it to them at its actual size.)

It has come to my attention that some installers have modified the standard document styles. **THIS IS STRICTLY FORBIDDEN.** The only changes to these styles that should be made are those necessitated by the use of different fonts. If you don't have a font that's called for in the standard style, do the best you can. If this produces noticeably different results, mention the difference in the *Local Guide*. Users expect the standard styles to produce the same output at different sites. If you must create local styles, give them different names and describe them in the *Local Guide*. The new manual describes what happens when `SAMPLE.TEX` is run with some modifications. Users will be unhappy if changes to the document style produce different results than is claimed in the book.

Speaking of document styles... before creating a document style for anyone else to use, talk to a typographic designer. People with no training in design who do their own formatting invariably do a rotten job. This is discussed in the new manual.

Enjoy.



A Solution to the Tower of Hanoi Problem Using TEX

Bruce Leban

Here is a solution to the classic Tower of Hanoi problem using TEX. This solution actually produces a printed solution to the problem illustrating the states of the stacks at each stage. Examination of this program may be instructive in understanding the operations of TEX's macro packages.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% \hanoi
%
% The basic macro that solves the Tower of Hanoi problem is called \hanoi.
% The first argument is the number of disks and the second is a list of disks.
% Each disk is identified by a single digit from 2 to 9 denoting its size.

\def\hanoi#1#2{%
   \numdisks=#1%
   \gdef\1{#2}\gdef\2{}\gdef\3{}%
   \showtowers123%
   \solve123%
   \vfill\eject
   }

\newcount\numdisks

% \solve#1#2#3 ::  move from #1 to #3 using #2

\def\solve#1#2#3{%
   \ifnum \numdisks=1 %
       \move#1#3%
   \else
       {\advance\numdisks by -1 %
        \solve#1#3#2}%
       \move#1#3%
       {\advance\numdisks by -1 %
        \solve#2#1#3}%
    \fi}

% \move #1#2 ::  Move from #1 to #2

\def\move#1#2{%
   \line{\bf Move from #1 to #2:  \hfill}
   \message{Move from #1 to #2.  }
   \first{#1} \append{.}{#2} \gstore{#2}
   \rest{#1} \gstore{#1}
   \showtowers123%
   }
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Lisp like functions for TeX.
%
% In order to implement the tower of hanoi, we implement a small list
% processing system in TeX.  Lists are implemented as strings of characters
% (tokens) stored in a macro.    Each variable is stored in a macro of the
% corresponding name.    For example, the variable 'x' is stored in the macro
% '\x'.    Since it is convenient to pass around values directly, each function
% puts its result into the special variable '.'  (i,e., '\.').  For example, the
% Lisp code:
%     (setq a (append (first b) (rest c))
% would be coded as:
%     \first{b}          '.'  is now (first b)
%     \store{x}          'x' is now (first b)
%     \rest{c}           '.'  is now (rest c)
%     \append{x}{.}      '.'  is now (append (first b) (rest c))
%     \store{a}          'a' is now (append (first b) (rest c))
% The functions only support single-level lists (of tokens) and the function
% \first which produces the first element really produces the list of the first
% element, since these have the same representation.

% \value x  ::  \let\.=\x
% \Value x  ::  \let\:=\x  % We can use this to avoid clobbering \.
% \store x  ::  \let\x=\.
% \gstore x ::  \global\let\x=\.

\def\value #1{\expandafter\xvalue\csname#1\endcsname}
\def\xvalue{\let\.=}
\def\Value #1{\expandafter\xValue\csname#1\endcsname}
\def\xValue{\let\:=}

\def\gstore #1{\expandafter\xgstore\csname#1\endcsname=\.}
\def\xgstore{\global\let}
\def\store #1{\expandafter\let\csname#1\endcsname=\.}

% \append #1#2 ::  \.  <== (append #1 #2)

\def\append #1#2{\Value{#1}
                 \value{#2}
                 \edef\.{\:\.}}

% \first #1 ::  \.  <== (list (first #1))

\def\first #1{\value{#1}\expandafter\xfirst\.?!}
\def\xfirst #1#2!{\edef\.{#1}}

% \rest #1 ::  \.  <== (rest #1)

\def\rest #1{\value{#1}\expandafter\xrest\.??????????????????!}
\def\xrest #1#2?#3!{\edef\.{#2}}
```

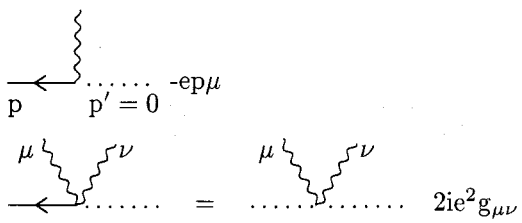
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% These functions are what actually display the towers.

\def\showname#1#2{
   \hbox to \hsize{%
      \hskip #2%
      \hbox to \towerwide{%
          \hfill {\bf #1}\hfill}%
      \hfill}}

\def\showdisk#1#2{%
   \hbox to \hsize{%
      \hskip #2%
      \hbox to \towerwide{%
          \hfill
          \vbox {\hrule height \diskhigh width #1\diskwide}%
          \hfill}%
      \hfill}%
   \vskip\diskvskip}

\def\showdisks#1#2.#3{%
   \if #1/
   \else \showdisk#1{#3} \showdisks#2.{#3}\fi}

\def\showtower#1/#2#3{%
   {\vbox to \towerhigh{%
      \vfill
      \showdisks#1/.{#3}
      \showname{#2}{#3}}}}

\def\showtowers#1#2#3{%
   \medskip
   \value{#1}
   \expandafter\showtower\./A{0pt}%
   \nointerlineskip
   \nobreak\vskip -1\towerhigh
   \value{#2}
   \expandafter\showtower\./B{1.05\towerwide}%
   \nointerlineskip
   \nobreak\vskip -1\towerhigh
   \value{#3}
   \expandafter\showtower\./C{2.1\towerwide}%
   \bigskip\goodbreak}

\baselineskip=0pt
\newdimen\diskwide\diskwide=9pt
\newdimen\diskhigh\diskhigh=5pt
\newdimen\diskvskip\diskvskip=3pt          % Vertical spacing between disks.
\newdimen\towerwide\towerwide=5\diskwide   % This is >= largest disk number.
\newdimen\towerhigh\towerhigh=5\diskhigh   % This is > number of disks.
           \advance\towerhigh 5\diskvskip

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% And now prove it all actually works.

\hanoi3{234}
```

Wiggly Lines

A. G. W. Cameron
Harvard University

During the summer of 1985 the IBM PC Technical Group of the Boston Computer Society carried out a project to test scientific word processing programs. As part of the project the reviewers were given a challenging set of benchmarks which were to be implemented if possible on the various packages. In addition to various standard mathematical constructs which TEX can handle with ease, the benchmarks contained an electrical circuit diagram, a Feynman diagram (used in high energy physics), and organic chemical structural formulas. I set out to demonstrate that plain TEX could do all of these, and I succeeded apart from having to make a few minor font substitutions.

Two of the special benchmarks which I mentioned above involved drawing diagonal lines, and the Feynman diagram involved wiggly lines, one vertical and several diagonal. The following shows the essential part of this benchmark.



The method by which I produced slanted lines in the circuit diagram and the chemical structural formulas is a simplification of the method used with the above wiggly lines, so I give here only the technique for doing the wiggly lines. When I started I was unaware of the procedure for drawing diagonal lines due to Amy Hendrickson (TUGboat, Vol. 6, No. 2, p. 83). Nevertheless, my initial trials used a technique similar to hers, laying down overlapping dots. However, since I was developing the method using an Epson FX-80 printer, which produces large and fuzzy dots, I began to worry about the different dot sizes that would occur with other output devices, and also about positional uncertainties in laying down the dots.

Therefore I used square rules. These were made a half point on the side in the above example. For the circuit diagram and the chemical structures I was forced to go to squares one point on the side in order to stay within the available TEX memory.

The procedure involved using TEX as a plotting device. Suppose the current position is (x_0, y_0) and that we wish to place a square at the relative coordinates (x, y). Suppose also that (x, y) is at the center of the left side of the square that is to be plotted. Then I would use an `\hskip` for the distance x. I would then describe the rule as a `\vrule height` $(y + 0.5h)$ `width` h `depth` $(-y + 0.5h)$, where h is the length of the side of the square rule. Then I would go back to the origin with an `\hskip` for the distance $-(x + h)$. It may readily be seen that this procedure uses two `\hskip`'s where one could be made to do in repeated plotting of points, but it simplifies the calculations.

To get a vertical wiggly line as in the above example one simply sets (x_0, y_0) to the base of the wiggly line and integrates the equations $\dot{y} = c_1$ and $\dot{x} = -c_2 x$ using the `\loop...\repeat` structure, where c_1 and c_2 are constants. To get the slanted wiggly lines in the above diagram one just applies a rotation of the coordinate system to the above solution so that new points (x', y') are plotted. To illustrate all of this I give below the code to produce a wiggly line slanted up to the right.

```
% define 12 variables:
\newcount\vone ...\newcount\vtwelve
% set up initial conditions
$\vone=0\vtwo=0\vthree=7000\vfive=\vtwo
\loop % begin the iteration; assume that we
% have an x (\vtwo) and y (\vone) and rotate
% the coordinates to obtain x' and y'.    I take
% the angle of rotation to be 30°, so that
% sin 30° = 0.5 and cos 30° = 0.866.
\vseven=\vfive\multiply\vseven by 866
\divide\vseven by 1000
\vnine=\vone\divide\vnine by 2
\advance\vnine by \vseven
\vsix=\vone\divide\vsix by 10
% here we avoid an overflow
\multiply\vsix by 866\divide\vsix by 100
\veight=\vfive\divide\veight by 2
\multiply\veight by -1
\advance\veight by \vsix
% now \veight is y' and \vnine is x'
\vten=16384\advance\vten by \veight
\veleven=16384\advance\veleven by -\veight
\hskip\vnine sp %move x' and raise rule by y'
\vrule height\vten sp width32768sp
       depth\veleven sp
\vtwelve=\vnine\advance\vtwelve by 32768
\hskip-\vtwelve sp
% if not done, increment y, then x
\ifnum\vone<1800000\advance\vone by 20000
\advance\vtwo by\vthree\vfour=-\vtwo
\divide\vfour by 10\advance\vthree by\vfour
\advance\vfive by \vtwo\repeat$
```

Stream lists and related list types for LaTeX

Reinhard Wonneberger*
Universität Hamburg

ABSTRACT. As a first step towards a more general concept of lists in LaTeX, 'stream lists' are introduced. While sharing explicit labeling with normal lists, they are set without linebreaks.

On Lists in General. Lists in general have the following advantages: 1. they help with organizing thought; 2. they provide ordered access to a group of items; 3. lists are especially useful with automatic formatting: (a) items can be easily accessed in the file by find or point commands; (b) producing text is supported in the following ways: i. automatic counting helps with modifications; ii. items may be referred to by crossreference tags; iii. the appearance of lists can be changed without touching the items; iv. different list styles can be chosen for proofreading and the final draft.

Normally the internal structure of lists is represented by linebreaks and indents.[1] Since this style improves overall orientation and access at the cost of consuming space and leaving ugly white zones on the page, it is closely related with the field of technical documentation.

In other areas, however, it may be preferable to have *stream lists.* An example of `enumerated` nested stream lists is given in the previous paragraph. While stream lists also represent the logical structure of the list through their explicit label information, they avoid interrupts of the flow of reading and improve the aesthetic appearance of the text.

Closely related to stream lists are *empty lists*, which do not show up in print; producing unlabeled running text from a list environment, they allow the user to keep 'hidden' list structures in his source text.

Looking beyond the scope of this paper, users like me, apart from the possibility of changing individual parameters, should be pleased to have at their disposal, also in LaTeX, the following list types on a *keyword* basis: 1. *spaced*, which is now the default; 2. *compact*, which is like *spaced*, but with normal `\baselineskip` also between items; 3. *stream* and 4. *empty* both described in this article; 5. *par* placing the label at the beginning of an indented paragraph and 6. *hang* doing the same for paragraphs with hanging indentation.

Though a first step is done with the macros presented here, implementing such a general scheme should rather be a challenge for the wizards than a brave but hopeless endeavor for an amateur.

Introducing Stream Lists. Streamlists are derived here from the LaTeX list macros.[2] Though in the standard list macros much can be done by changing parameters in the second argument, streaming is impossible, since `\everypar` is used.

There is a `streamlist`–environment which corresponds to the normal `list`–environment. This environment may be used to specify labeling explicitly, e.g.

```
\newcounter{list}
\begin{streamlist}{(\alph{list})}
          {\usecounter{list}}
\item First element on this level.
\item Second element on this level.
\end{streamlist}.
```

It is also used implicitly by the `streamenumerate`–environment, which corresponds to the normal `enumerate`–environment and can be nested in the known way:

```
\begin{streamenumerate} ...
      \end{streamenumerate}
```

In normal lists, the skip before and after the item label is controlled by the *dimension parameters* `\itemindent` and `\labelsep` resp. Here we use two *commands* instead: `\labelpref` for the label prefix, which is initialized to `\quad`, normally an `em`-space, and `\labelsuff` for the label suffix, which is initialized to `\enspace`, normally an `en`-space. Since both of them are commands, they may also be used to surround labels with generated punctuation like square brackets, colons or the like.

---

* *Alttestamentliches Seminar, Sedanstr.19, D 2000 Hamburg 13, FRG.* The macros presented here were developed at *DESY, Notkestr.85, D 2000 Hamburg 54, FRG,* with the aid of P. K. SCHILLING. Comments are welcome to him (R02SCH) or the author (B03WBG) at DHHDESY3 via EARN (which is connected to BITNET).

[1] Correspondence of the logical structure of the text and its graphic representation has a long history of its own and is by no means trivial; in some cases, a lot of linguistic theory is involved. On this subject cf. R. WONNEBERGER: *Normaltext und Normalsynopse. Neue Wege bei der Darstellung alttestamentlicher Texte.* Zeitschrift für Sprachwissenschaft 3 (1984) 203-233.

[2] L. LAMPORT: The LaTeX Document Preparation System. Reading, Massachusetts / Menlo Park, California / London / Amsterdam / Don Mills, Ontario / Sydney: Addison-Wesley (forthcoming).

Macro-Definitions. Stream and empty lists are obtained with the aid of the following definitions:

```
%       \labelpref      :  macro executed right BEFORE an item label.
%       \labelsuff      :  macro executed right AFTER  an item label.

\def\streamlist#1#2%
  {\ifnum \@listdepth >5\relax \@toodeep
       \else \global\advance\@listdepth\@ne \fi
  \csname @list\romannumeral\the\@listdepth\endcsname
  \def\labelpref{\quad}\def\labelsuff{\enspace}
  \def\@itemlabel{#1}\let\makelabel\@stlab \@nmbrlistfalse #2\relax
  \let\@item=\@streamitem
  \ignorespaces}

\def\@stlab#1{{#1}}

\def\@streamitem[#1]{\if@nmbrlist \refstepcounter{\@listctr}\fi
  \labelpref{\makelabel{#1}}\nobreak \labelsuff \ignorespaces}

\def\endstreamlist{\global\advance\@listdepth\m@ne}

\def\streamenumerate{\ifnum \@enumdepth >3 \@toodeep\else
  \advance\@enumdepth \@ne
  \edef\@enumctr{enum\romannumeral\the\@enumdepth}\streamlist
    {\csname label\@enumctr\endcsname}{\usecounter
      {\@enumctr}\def\makelabel##1{{##1}}}\fi}

\let\endstreamenumerate =\endstreamlist

\def\streamempty{\ifnum \@enumdepth >3 \@toodeep\else
  \advance\@enumdepth \@ne
  \streamlist{\relax}{\def\makelabel##1{\relax}\let
  \labelpref=\relax
  \let\labelsuff=\relax}\fi}

\let\endstreamempty =\endstreamlist
```

LaTeX Command Summary Available from TUG

L. Botway and Chris Biemesderfer, of the Space Telescope Science Institute, Johns Hopkins University, have prepared a 15-page LaTeX Command Summary for use with the LaTeX installation at the Institute (TeX Version 1.2, LaTeX Version 2.05). This document has been made available to TUG for wider distribution. When ordering, specify "LaTeX command summary"; the price will be $5.00, to cover duplication, shipping and handling.

Sample entries:

`\'` in `tabbing` environment moves current column to the right of the previous column [L102]. Elsewhere, acute accent, as ó [T52].

`\indexentry{text}{ref}` is written to the `.idx` file for `\index{text}` occurring at reference `ref` [L136].

`\leftrightarrow` is ↔. `\Leftrightarrow` is ⇔ (math mode) [T437,L50].

`\makebox[size][pos]{text}` creates a box of dimension `size` containing `text` at optional `[pos]` [L87]. In `picture` environment, `\makebox(width,height)[pos]{text}` puts `text` in a box; dimensions of box are `width` and `height`; `text` is positioned at optional `[pos]` (see `positions` on page 9) [L109].

Letters

Inquiry: Historical compilations

To the Editor:

[O]ur project is a historical editing enterprise. We are interested in contacting other TeX users who might be doing applications of a similar nature to ours.

Rick Hendricks
Assistant Editor
The Vargas Project
Zimmerman Library
The University of New Mexico
Albuquerque, NM 87131

Editor's note: Two other such projects are known to the Editor: at Princeton University's Firestone Library, the Papers of Thomas Jefferson; and at Stanford University, the Collected Works of Kurt Gödel. The latter will be published by Oxford University Press. Volumes I and II will contain all of Gödel's previously published works (articles originally in German will be accompanied by English translations), and succeeding volumes will contain unpublished manuscripts, lectures, lecture notes, correspondence, and extracts from scientific notebooks. Volume I is scheduled for publication at the end of 1985. TeX contacts for the Gödel project are: Yasuko Kitajima, Aldine Press,
12625 La Cresta Drive,
Los Altos Hills, CA 94022
and Carolyn Talcott,
Department of Computer Science,
Stanford University, Stanford, CA 94305
(CLT@SU-AI.ARPA)

Inquiry: 35mm Slides with SliTeX

To the Editor:

We would like to get 35mm slide output from SliTeX. Has anyone tried to do this?

Right now, we are looking at a Matrix QCR film recorder and Lasergrafix rasterizing box. Do you know of any drivers that work with the Lasergrafix?

Any other suggestions?

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

News & Announcements

CALENDAR

1985

Harvard University, Cambridge, Massachusetts

Nov. 18–20 Beginning TEX

Nov. 21–22 Intermediate TEX

Dec. 9–11 Advanced TEX

Dec. 12–13 TEX macro writing

Addison-Wesley, Reading, Mass.
TEX on the IBM-PC

For technical IBM-PC users

Dec. 2–3, 7:00–10:00 p.m.
Dec. 7 9:00 a.m.–3:00 p.m.

For non-technical IBM-PC users

Dec. 5–6, 1:00–4:00 p.m.
Dec. 14 9:00 a.m.–3:00 p.m.

* * * * *

DECUS, Anaheim, California

Dec. 8 LATEX — A sophisticated text formatting language (Leslie Lamport)

Dec. 9 TEX for the DECUS Proceedings (Barbara Beeton) 1:00–2:00 p.m.; Marina Room 4

* * * * *

1986

Joint Mathematics Meetings, New Orleans, Louisiana

Jan. 9 TEX presentation (Bart Childs) 7:30–8:30 p.m.

Jan. 7–11 TEX exhibits

Jan. 11 TEX seminar (Bart Childs) 9:00 a.m.–3:00 p.m.; Desiree Room, Days Inn, Canal Street

* * * * *

Jan. 15 TUGboat Volume 7, No. 1: Deadline for submission of manuscripts

Feb. 1 Deadline for submission of items for the preliminary program, TEX Users Group Annual Meeting

Apr. 14–16 International Conference on Text Processing and Document Manipulation, University of Nottingham, England

Apr. 28 TUGboat Volume 7, No. 2: Deadline for submission of manuscripts (tentative)

June 19–21 TEX for Scientific Documentation, Strasbourg, France (see announcement, p. 160)

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.

Acknowledgment of Contributions

The Officers and Steering Committee gratefully acknowledge receipt of royalties and other contributions to TUG from several sources during 1985:

1. From the sale of Don Knuth's *TEXbook*, royalties of $3,787.
2. From David Kellerman and Barry Smith, of Kellerman & Smith, royalties of $1,325, from fees paid to them for distribution of the VAX/VMS version of the WEB sources of TEX.
3. More than 600 copies of Arthur Samuel's First Grade TEX sold by TUG in 1985 resulted in the addition of $4,000 to TUG's treasury.

TUG sincerely appreciates these very generous contributions.

Samuel B. Whidden, Treasurer

TEX FOR SCIENTIFIC DOCUMENTATION

Call for papers

Strasbourg, France, June 19–21, 1986

The second European conference on the TEX system and applications will be held June 19–21, 1985 at Strasbourg, France. TEX, a trademark of the American Mathematical Society, is the well-known composition system developed at Stanford University by Prof. D. Knuth. It is suitable for high-quality scientific and technical documentation.

The aim is to provide a state-of-the-art survey of current work in this area and to encourage technology transfer and information exchange on the latest applications of documentation systems based on TEX.

Suggested presentation topics are:

- TEX system integration and improvement
- Typeface design
- TEX macro packages and development methodology
- Document structure standards
- User interfaces and workstations
- Previewing, printing and typesetting
- Implementation of output drivers
- Integration of text and graphics
- Filing, retrieval and delivery of TEX documents
- Using other page description languages with TEX
- Document preparation environments
- Electronic publishing applications

Programme Committee Members:

J. André	IRISA Rennes (France)
B. Beeton	Amer. Math. Soc. (U.S.A.)
H. Brown	Univ. Kent (G.B.)
G. Degli Antoni	Univ. Milano (Italy)
D. Feingold	EDF Clamart (France)
D. Knuth	Stanford Univ. (U.S.A.)
D. Lucarella	Univ. Milano (Italy)
P. MacKay	Univ. Washington (U.S.A)
J.-J. Quisquater	Philips Bruxelles (Belgium)
B. Schulze	Univ. Bonn (Germany)

Organizing Committee Members:

J. Désarménien	Univ. Strasbourg (France)
R. Goucher	Amer. Math. Soc. (U.S.A.)
D. Lucarella	Univ. Milano (Italy)

Please return the form below to

J. Désarménien
Laboratoire de typographie informatique
Université Louis-Pasteur
7, rue René-Descartes
67084 Strasbourg Cedex (France)

as soon as possible, and in any case **before 1 January 1986.**

Papers accepted for the conference will be collected and published.

Timetable:

1 March 1986	Submission of papers
15 April	Notification of acceptance
19–21 June	Conference at Strasbourg

To J. Désarménien, Laboratoire de typographie informatique, 7, rue René-Descartes, Université Louis-Pasteur, 67084 Strasbourg Cedex (France)

◯ I am interested in attending the conference. Please send me further information.
◯ I would like to submit a paper at the conference. Please send me a copy of the Instructions to Authors.

TUG Business

Proposed TUG 1986 Budget

	Budget 1986
Income:	
Membership/Subscriptions	
Individual/Library[1,2]	$ 22,000
Institutional Membership[1,2]	
Educational	9,000
Non-educational	11,500
Publications/Merchandise	
Back issue sales[3]	8,000
Publications/merchandise for resale	6,000
Other publications[4]	8,000
Meetings[2,5]	
Annual meeting	12,000
Other meetings/courses	180,000
Exhibitor fees	1,500
Other	
Videotape sales/rental	7,000
Advertising/mailing list sales	4,500
Royalties/contributions[6]	5,000
Total income	$ 274,500
Expenses:	
TUGboat (3 issues)	
Printing	$ 10,000
Postage	2,500
Editorial services	9,000
Computer expenses	4,500
Meetings*[,2,5]	
Annual meeting	8,000
Other meetings/courses	70,000
Other	
Other publications[4]	2,000
Resale of publications/merchandise*	5,000
Royalties*	500
ANSI meetings*	-0-
Legal and tax consulting*	1,500
Postage, general mailings	10,000
Printing back issues[3]	4,000
Printing, other	3,000
Personnel	70,000
Technical assistant*[,7]	20,000
Admin/benefit costs*	1,600
Subsidies*[,8]	-0-
Video tape duplication*	3,000
Computer expenses	18,000
Programming[9]	6,500
Stanford TEX Project Support*[,10]	18,800
Miscellaneous*[,11]	6,000
Total expenses	$ 273,900

Budget summary:	
Total income	$ 274,500
Total expense	273,900
Net income/expense	600
Balance forward	55,789
Ending balance	$ 56,389

Notes: Expense figures, except those with an asterisk (*), include an 20% overhead charge for services rendered by the American Mathematical Society.

1. Current 1985 membership figures are: 1,440 members/subscribers, including 96 Institutional Members: 47 educational; 49 non-educational.
2. Advertising of TUG and the TUG Meetings and Courses will be accomplished via news releases to various trade publications, as well as by direct mailings to members and former members.
3. Back issues continue in great demand. Owing to this popularity, new-issue print orders are set high enough to accommodate expected back issue sales for 3 years.
4. The popularity of Arthur Samuel's "First Grade TEX" is expected to generate a large portion of this income; however, other publications are expected to be made available through TUG in the future.
5. The Annual Meeting registration fee was reduced substantially for the 1985, resulting in a significant reduction in meeting receipts. However, the number of attendees was lower than in 1984, so the anticipated effect that lowered fees would increase attendance was at best inconclusive. The 1986 meeting will be held at Tufts University, Medford, Mass. It is certain that expenses at Tufts will be higher than at Stanford, therefore it will be necessary to adjust fees accordingly.
   In addition, two- and three-day TEX courses are being planned at Illinois Institute of Technology, Harvard, Rutgers and Vanderbilt Universities, and other locations. A schedule will be announced in December.
6. Don Knuth designated a portion of the royalties from sales of *The TEXbook* to be donated to TUG. Contributions are anticipated from other sources.
7. The addition of a technical assistant to the staff is tentatively planned for midyear, conditioned upon the availability of funds.
8. This amount is available to the Finance Committee to subsidize travel and membership/participation fees for individuals when appropriate.
9. Improving the functioning of the TUG data base.
10. The Steering Committee approved the expenditure of $25,000 in support of the TeX project at Stanford to be paid over a 12-month period commencing October 1, 1985.
11. Postage/express charges, telephone tolls and supplies, plus programmer and clerical services not associated with production of TUGboat.

Respectfully submitted,
Samuel B. Whidden, Treasurer

Laser printer drivers available for MicroTEX™!

MicroTEX can now be used with a screen previewer!

In response to requests from MicroTEX users, Addison-Wesley will distribute a number of products from Textset, Inc. of Ann Arbor, Michigan. These products will include TEX™ Preview as well as a number of laser printer drivers, including PostScript™, Imagen™ and QMS Lasergraphix™ series printers.

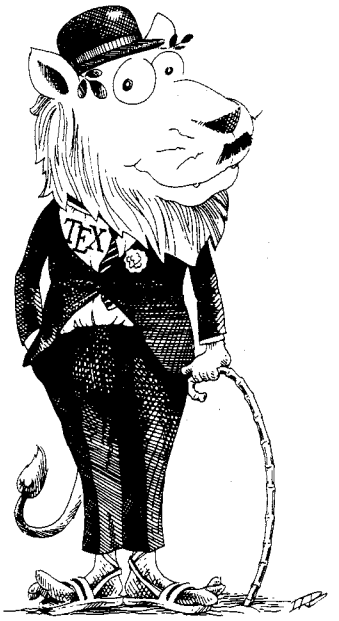
Configurations needed for screen previewer:
* HERCULES Graphics Card™ * IBM™ PC/XT or AT

The Postscript driver makes it possible to use MicroTEX with the following devices:

* Apple LaserWriter™
* QMS 800 and 2400
* Diconix Dijit 2
* Apollo DOMAIN/Laser26
* Sun LaserWriter

In addition to the driver for the IBM and Epson dot matrix printers bundled with MicroTEX, Addison-Wesley now has available a driver for the Toshiba™ dot matrix printer. The speed of the Toshiba printer makes it ideal for small-scale technical projects.

Site licenses for MicroTEX and all Educational Media Systems Division products available.
Call for further information.



This was produced with MicroTEX and an Apple LaserWriter with a PostScript driver.

MicroTEX Order Form

☐ Yes, please send me:

☐ MicroTEX	$495	☐ Imagen Driver	$200
☐ Toshiba Driver	100	☐ QMS Lasergraphix Driver	200
☐ PostScript Driver	300	☐ TEX Preview	250

☐ My check is enclosed (Orders paid by check will be shipped free of charge.)
☐ Visa ☐ MasterCard (Interbank#_____) ☐ American Express

Card No._____ Exp. Date_____

Signature_____

☐ Please bill my company. Your company order form and P.O. number must accompany your order. Thank you.

Name_____ Title_____

Firm/Institution_____ Department_____

Street_____ City/State/Zip_____

Telephone(____) _____ Ext._____ Best time to call_____

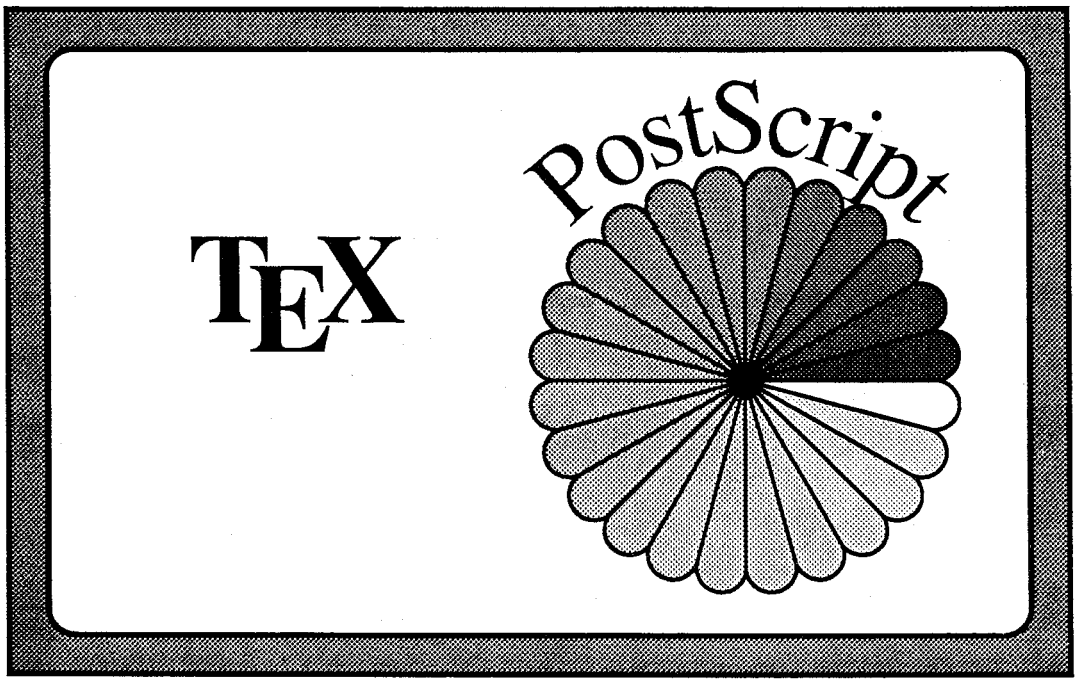
☐ I am interested in MicroTEX, but have questions. Please call me.
☐ Please keep me informed about all MicroTEX developments and related products. 35184

Send your order to:

Addison-Wesley Publishing Company
Educational Media Systems Division
Reading, MA 01867
(617) 944-3700 ext. 2677

TEXTSET, Inc. 416 Fourth Street • P.O. Box 7993 • Ann Arbor • MI • 48107 • (313) 996-3566
technical composition software and services

Complete TEX and PostScript Support



Announcing New VAX/VMS Software

TEXTSET's TEX Preview is being ported to the VAXstation II with the cooperation of Digital Equipment Corporation. DVILASER/PS, our TEX-to-PostScript device driver software is now available for VAX/VMS.

Sun and Apollo Support

TEXTSET's TEX Package, TEX Preview, and DVILASER/PS are available for Sun and Apollo workstations. DVILASER/PS supports any PostScript printer including the Sun LaserWriter and the Apollo DOMAIN Laser/26.

PostScript, DVILASER, and TEX, when used in combination can produce some very interesting output.

Ask Addison-Wesley Publishing Co., Inc. or Personal TEX, Inc. about **TEXTSET***'s TEX Preview and DVILASER/PS for 8086 family MSDOS microcomputers.*

TEX Lectures on Videotape

Videotapes of lectures from a series of 2-day short courses, ranging from 10 to 12 hours each, and some lectures presented at TEX Users Group meetings are available for lease or purchase in VHS, Beta or $\frac{3}{4}''$ U-Matic formats. For all videotapes except those of the 1984 TUG meeting, prices for lease or purchase are listed at the bottom of the next page.

For additional information, contact Ray Goucher, TEX Users Group, P. O. Box 9506, Providence, RI 02940-9506, U.S.A., (401) 272-9500, ext. 232.

The Elements of METAFONT Design – Georgia Tobin

This short course was presented in conjunction with the August 1985 TEX Users Group Meeting. It was designed to provide an introduction to font design using METAFONT, giving the student an indepth understanding of the nature and structure of a font of type, an easy familiarity with the concept of a meta-font of type, and a nodding acquaintance with the grammar and syntax of METAFONT. Topics covered in detail are: the basics of digital type design; the concept of a METAFONT; the basic structure of meta-code; points, pens and paths; macros; conditions and the use of conditional code; and testing fonts. Course exercises ranged from discussion of design philosophy and approaches to the coding tasks to actual writing of METAFONT code.

Necessary prerequisites which may strike some as unsavory (e.g., vector arithmetic, basic trigonometry, tidy programming habits) were covered as warranted by the expertise of course participants.

Georgia Tobin is with The Metafoundry, OCLC Online Computer Library Center, Inc., Dublin, Ohio.

Traditional Typography Meets LATEX – Marshall Henrichs

In the course of reviewing and working out design standards to be incorporated in the manual to Leslie Lamport's macro package LATEX, Marshall Hendrichs, Art Director for Addison-Wesley Publishing Co., was forced to evaluate the task at hand in the same light that typographers have had to evaluate all technological innovations throughout four centuries of typographic evolution. This course was developed from his experience. In it he presents a short history of the evolution of traditional typographic standards and addresses such issues as: Can the reader understand what has been written? What elements lead to understanding and what elements impede understanding? How much effort is needed to overcome the impediments? What can be read in today's newspapers and textbooks? What is allusive typography and what role does it play in comprehension? When does humor and analogy confuse the reader? When form and content get divorced, who gets the house?

This course does not duplicate the one described below: "First Principles of Typographic Design for Document Production". It offers a different perspective on key issues in document design (e.g., legibility and aesthetics).

Book Design Utilizing TEX – Richard Southall and Leslie Lamport

At the August 1984 TUG meeting a short course, "First Principles of Typographic Design for Document Production", was presented. The course establishes some basic principles for the typographic design of simple text. The application of these principles to the design of documents, and the implementation of the resulting designs with TEX, was discussed. Topics include: typographic structures in text; LATEX structures; graphic conventions in text; the document designer's tools; making text readable; designing headings; implementing text and heading designs; designing pages; implementing page layouts; list design and other issues.

Introduction to the Internal Workings of TEX82 – Donald Knuth

In conjunction with the July 1982 TEX Users Group meeting a short course, "Introduction to TEX82", was presented on the internal workings of TEX82. The WEB source of TEX82 was used as a reference. A reading knowledge of PASCAL was strongly recommended as a prerequisite.

The following topics were covered: reading WEB programs; representation of strings; data structures for boxes and glue; representation of control sequences; syntactic routines (TEX's eyes and mouth); semantic routines (TEX's stomach and intestines); breaking paragraphs into lines; hyphenation; scanning file names; input of font metric (TFM) files; output of device-independent (DVI) files; initializing a TEX production program.

The principal goal of the course was to make the participant familiar with the anatomy of the TEX82 system, so that it will be clear how to make system dependent changes necessary to install it as an effective production tool. Considerations necessary for tailoring TEX for use with languages other than English were discussed. The information provided by this course should make the viewer able to make better use of TEX than would otherwise be possible and to help with troubleshooting when others at the site do strange things with the software.

Introduction to AMS-TEX82 – Michael Spivak

At the July 1983 TEX Users Group meeting an Introductory AMS-TEX82 Users Course for secretaries and technical typists was presented by Michael Spivak, author of *The Joy of TEX*, which served as the text for this course. The course introduced the use of the macro package AMS-TEX, as revised for TEX82, which concentrates mainly on the problems of typesetting complicated mathematics; it is not a course in how to use TEX itself. The concepts presented, however, are applicable to other macro packages based on TEX82. (Document preparation with TEX is usually approached through a suitable macro package rather than directly through plain TEX.) In order to derive maximum benefit from the use of these videotapes, the user should be familiar with the use of a text editor and at least somewhat familiar with the demands of technical typing.

TEX Users Group Meeting, Stanford University, August 15–17, 1984

The following sessions were videotaped and are available for lease ($50 each hour or fraction thereof) or purchase ($75 each hour or fraction thereof):

Pierre MacKay: Introduction to TEX and TUG for new users (1:00); Don Knuth: Update on TEX82 and general Q & A (1:30); Jacques Désarménien: Running TEX in a French enviroment (fonts, hyphenation, typography) (1:30); Don Knuth et al.: Macro writing (3:00) and What's new in METAFONT and typography (1:30); Site Coordinators' progress reports – DG MV 8000, Prime 750, HP 3000, IBM Group, UNIX, and VAX (VMS) (1:30); David Fuchs, John Gourlay and Peter Sih: Output devices and drivers (1:00); Leslie Lamport and Richard Southall: Interfacing conventional design practice with TEX and METAFONT: discussion (1:30); and Georgia Tobin: Font design using METAFONT & discussion (1:00).

Videotape Lease/Purchase Prices

	Lease	Purchase*
List	$425/month	$800
Institutional member, non-educational	375/month	700
Educational institution	350/month	650

*The **purchase price** for VHS and Beta formats is $100 less in each category.

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, $15.00 per issue, circle issue(s) desired: 1980 (v. 1) #1; 1981 (v. 2) #1, #2, #3; 1982 (v. 3) #1, #2; 1983 (v. 4) #1, #2; 1984 (v. 5) #1, #2; 1985 (v. 6) #1, #2, #3	

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

* If your computer is "software compatible" with another type (e.g. Amdahl with IBM 370), indicate the type here.

Shipping and Handling

Books are sent via surface mail (UPS to U.S. addresses and printed matter elsewhere) unless air delivery is requested. Shipping and handling charges for book orders are shown below. These charges apply only to books indicated by an asterisk (*).

	First Book	Each Additional	Maximum
Surface	$2	$1	$ 25
Air	$5	$3	$100

- *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.

Ordered by: _____ Mail to (if different): _____

QTY	ITEM	AMOUNT
	TUGboat back issues, $15.00 per issue, circle issue(s) desired: 1980 (v. 1) #1; 1981 (v. 2) #1, #2, #3; 1982 (v. 3) #1, #2; 1983 (v. 4) #1, #2; 1984 (v. 5) #1, #2; 1985 (v. 6) #1, #2, #3	
	The TEXbook by Donald E. Knuth – $17.00* each (with Errata and Changes, add $2.00 each)	
	The TEXbook: Errata and Changes (included with TUGboat) – additional copies $3.00 each	
	TEX: The Program by Donald E. Knuth – $25.00* each	
	First Grade TEX: A Beginner's TEX Manual by Arthur L. Samuel – $6.00 each	
	The Joy of TEX (rev. prelim. ed., 1982, with AMS-TEX82 suppl.) by M. Spivak – $10.00* each	
	LATEX: A Document Preparation System by Leslie Lamport – $19.00* each	
	User's Guide to the HP TEX Macros by Susan Daniels – $6.00 each	
	TEX and Metafont: Errata and Changes (TEX78 final edition, September 1983) – $4.00 each	
	Mathematics into Type by Ellen Swanson – $15.00* each	
	Proc. First European Conf. TEX Scientific Documentation, D. Lucarella, Editor – $40.00* each	
	TEX Lectures on Tape (see TUGboat Vol. 6, No. 3 for complete listing)	
	Shipping and handling Surface ☐ Air ☐	

Quantity discounts available on request.
Prices are subject to change without notice.

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

Brief descriptions of TEX and TEX-related publications

TUGboat

TUGboat is the newsletter of the TEX Users Group. It contains communications from the Stanford TEX Project; articles of interest to installers and users of TEX and METAFONT; reports of activity at distribution centers and user sites; macros, problems, questions and answers; a calendar of TEX and TUG-related events; and official TUG business.

Three issues will be published in 1986. Memberships and subscriptions are accepted on a calendar-year basis only; for information, write to TUG at the address given above for general correspondence, or call (401) 272-9500, ext. 232. Back issues are available as indicated above.

(*Continued*)

The TEXbook by Donald E. Knuth

From the back cover of *The TEXbook*: "TEX represents the state-of-the-art in computer typesetting. It is particularly valuable where the document, article, or book to be produced contains a lot of mathematics, and where the user is concerned about typographic quality. TEX software offers both writers and publishers the opportunity to produce technical text, in an attractive form, with the speed and efficiency of a computer system.

"Novice and expert users alike will gain from *The TEXbook* the level of information they seek. Knuth warns newcomers away from more difficult areas, while he entices experienced users with new challenges. The novice need not learn much about TEX to prepare a simple manuscript with it. But for the preparation of more complex documents, *The TEXbook* contains all the detail required."

Published by Addison-Wesley Publishing Co., Inc., Reading, Mass., 1985.

TEX: The Program by Donald E. Knuth

From the introduction: "The main purpose of the ... program is to explain the algorithms of TEX as clearly as possible. As a result, the program will not necessarily be very efficient when a particular Pascal compiler has translated it into a particular machine language. However, the program has been written so that it can be tuned to run efficiently in a wide variety of operating environments by making comparatively few changes. Such flexibility is possible because the documentation ... is written in the WEB language, which is at a higher level than Pascal; the preprocessing step that converts WEB to Pascal is able to introduce most of the necessary refinements. Semi-automatic translation to other languages is also feasible, because the program ... does not make extensive use of features that are peculiar to Pascal."

Published by Addison-Wesley Publishing Co., Inc., Reading, Mass., 1984.

First Grade TEX: A Beginner's TEX Manual
by Arthur L. Samuel

From the introduction: "This is an introductory ready-reference TEX82 manual for the beginner who would like to do First Grade TEX work. Only the most basic features of the TEX system are discussed in detail. Other features are summarized in an appendix and references are given to the more complete documentation available elsewhere."

The Joy of TEX by Michael Spivak

The Joy of TEX is the user's guide for AMS-TEX, an extension of Donald Knuth's typesetting program TEX. TEX itself was designed to typeset a wide range of technical material, while AMS-TEX and *The Joy of TEX* were written to allow for simplified input of specifically mathematical text to TEX. Use of AMS-TEX requires no expertise in mathematics or in computer science. This edition consists of two parts: a reprint of the previous edition, which was written for the first version of TEX (SAIL TEX or TEX78) and a supplement documenting the changes and new features available with the current version of TEX (TEX82, written in Pascal). Use of AMS-TEX requires no more than that TEX itself be available. Published by the American Mathematical Society, Providence, R.I., 1982.

LATEX – A Documentation Preparation System
by Leslie Lamport

From the preface: "The LATEX documentation preparation system is a special version of Donald Knuth's TEX programs. TEX is a sophisticated program designed to produce high-quality typesetting, especially for mathematical text. LATEX adds to TEX a collection of commands that simplify typesetting by letting the user concentrate on the structure of the text rather than on formatting commands. In turning TEX into LATEX, I have tried to convert a highly-tuned racing car into a comfortable family sedan. The family sedan isn't meant to go as fast as a racing car or be as exciting to drive, but it's comfortable and gets you to the grocery store with no fuss. However, the LATEX sedan has all the power of TEX hidden under its hood, and the more adventurous driver can do everything with it that he can with TEX."

Published by Addison-Wesley Publishing Co., Inc., Reading, Mass., 1985.

Proceedings of the First European Conference on TEX for Scientific Documentation
Como, Italy, May 16–17, 1985, Dario Lucarella, Editor

From the preface: "This volume collects the papers presented at the European Conference '*TEX for Scientific Documentation*' held at Villa Olmo, Como, in May '85. ... The aim of the Conference was to provide a state-of-the-art survey of current research activities and the latest applications that are growing around TEX.

"The topics covered in the selected papers concern the following fields: Documentation systems based on TEX; TEX as a tool for authors; Customization of TEX for non-English languages; Hyphenation; Standardization problems; Facilities for interactive entering and retrieving of formulae; METAFONT and font design; Implementation of TEX, METAFONT and drivers." All papers are in English.

Published by Addison-Wesley Publishing Co., Inc., Reading, Mass., 1985.

Mathematics into Type by Ellen Swanson

This book covers the publication of mathematics from manuscript to the printed book or journal, with emphasis on the preparation of the copy for the compositor and the proofreading and makeup of the publication. It will be useful to the author who is directly concerned in the editing of his book, and it should benefit any author who is preparing a manuscript for publication.

Published by the American Mathematical Society, Providence, R.I., 1982.

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.

Eight 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. You may request densities of 6250, 1600 or 800 (800 is discouraged since it is more trouble to make).

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. It is assumed that one set of manuals will suffice you. If you require more than two sets, please write for prices since we must ask for more money for postage and handling.

Please send a check or money order (payable 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. Please check this out before sending a purchase order since many large firms seem to be unable to make prompt payment (or don't worry about it).

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 some future 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** ** density (6250, 1600 or 800) = _____

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 (300 dots/inch)

_____ 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 (preliminary edition) – $8.00

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 North America: $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

3/85