TEX, of course, is not for the faint of heart....

*The Seybold Report on Publishing Systems* (September 28, 1987)

# TUGBOAT

THE TEX USERS GROUP NEWSLETTER

EDITOR BARBARA BEETON

VOLUME 8, NUMBER 3 • NOVEMBER, 1987

PROVIDENCE • RHODE ISLAND • U.S.A.

## TUGboat

The communications of the TEX Users Group are published irregularly at Providence, Rhode Island, and are distributed as a benefit of membership both to individual and institutional members. Three issues of TUGboat are planned for 1988.

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

## Submitting Items for Publication

The deadline for submitting items for Vol. 9, No. 1, is January 19, 1988; the issue will be mailed in March.

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

Contributions in electronic form are encouraged, via electronic mail, on magnetic tape or diskette, or transferred directly to the AMS computer; contributions in the form of camera copy are also accepted. For instructions, write or call Barbara Beeton.

## TUGboat Advertising and Mailing Lists

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

## Other TUG Publications

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

## TUGboat Editorial Committee



*See page 255 for addresses.*

# Addresses

**Note:** Unless otherwise specified, network addresses (shown in `typewriter` font) are on the Arpanet.

**Terry L. Anderson**
AT&T Bell Laboratories – Liberty Corners
24 Hill Street
Bernardsville, NJ 07920
201-580-4428

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

**Barbara Beeton**
American Mathematical Society
P. O. Box 6248
Providence, RI 02940
401-272-9500
bnb@Seed.AMS.com,
bnb@xx.lcs.MIT.Edu,
Beeton@Score.Stanford.Edu

**Jeffery Boes**
Instrument and Avionic Systems Division
Lear Siegler, Inc
4141 Eastern Ave, SE
Grand Rapids, MI 49508

**Malcolm Brown**
AIR/Systems Development
Sweet Hall 3rd floor
Stanford University
Stanford, CA 94301
415-723-1248
MBB@jessica.Stanford.Edu

**Lance Carnes**
c/o Personal TEX
12 Madrona Avenue
Mill Valley, CA 94941
415-388-8853

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

**Adrian F. Clark**
Analysis Group
British Aerospace
E-Block, Manor Rd
Hatfield
Hertfordshire AL10 9LL, England
070-72-62300 x8216

**Malcolm Clark**
Imperial College Computer Centre
Exhibition Road
London SW7 2BP, England
Janet: texline@uk.ac.ic.cc.vaxa

**Richard V. Clayton**
Apt G4
289 Mt Hope Ave
Dover, NJ 07801
clayton@thumper.UUCP

**Maria Code**
Data Processing Services
1371 Sydney Dr
Sunnyvale, CA 94087
408-735-8006

**John M. Crawford**
Computing Services Center
College of Business
Ohio State University
Columbus, OH 43210
614-292-1741
Crawford-J@Ohio-State
Bitnet: TS0135@OHSTVMA

**Jackie Damrau**
Dept of Math & Statistics
Univ of New Mexico
Albuquerque, NM 87131
505-277-4623
Bitnet: damrau@unmb
UUCP: damrau@ariel.unm.edu

**Allen R. Dyer**
2922 Wyman Parkway
Baltimore, MD 21211
301-243-0008 or 243-7283

**Maureen Eppstein**
Administrative Publications
Stanford University
Encina Hall, Room 200
Stanford, CA 94305
415-725-1717
as.mve@Forsythe.Stanford.Edu

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

**David Fuchs**
1775 Newell
Palo Alto, CA 94303
415-323-9436

**Richard Furuta**
Department of Computer Science
Univ of Maryland
College Park, MD 20742
301-454-1461
furuta@mimsy.umd.edu

**Alonzo M. Gariepy**
OAIP, 18th floor, Mowat Block
900 Bay Street
Toronto, Ontario M7A 1L2 Canada
416-965-8765

**Raymond E. Goucher**
TEX Users Group
P. O. Box 9506
Providence, RI 02940-9506
401-272-9500 x232
reg@Seed.AMS.com

**Dean Guenther**
Computer Service Center
Washington State University
Computer Science Building, Room 2144
Pullman, WA 99164-1220
509-335-0411
Bitnet: Guenther@WSUVM1

**Klaus Guntermann**
Technische Hochschule Darmstadt
Fachbereich Informatik
Institut für Theoretische Informatik
Alexanderstraße 24
D-6100 Darmstadt
Federal Republic Germany
Bitnet: XITIKGUN@DDATHD21

**Doug Henderson**
Division of Library Automation
Univ of California, Berkeley
186 University Hall
Berkeley, CA 94720
Bitnet: dlatex@ucbcmsa

**Alan Hoenig**
17 Bay Avenue
Huntington, NY 11743
516-385-0736

**Don Hosek**
Platt Campus Center
Harvey Mudd College
Claremont, CA 91711
Bitnet: dhosek@hmcvax

**Doris T. Hsia**
2630 Sierra Vista Ct
San Jose, CA 95116
415-723-8117

**Patrick D. Ion**
Mathematical Reviews
416 Fourth Street
P. O. Box 8604
Ann Arbor, MI 48107
313-996-5273
ion@Seed.AMS.com

**Helmut Jürgensen**
Dept of Computer Science
Univ of Western Ontario
London N6A 5B7, Ontario, Canada
519-661-3560
Bitnet: A5O5@UWOCC1
UUCP: helmut@deepthot

**Benjamin J. Kennedy**
Xybion Corporation
240 Cedar Knolls Rd
Cedar Knolls, NJ 07927-1195
201-538-5111

**Donald E. Knuth**
Department of Computer Science
Stanford University
Stanford, CA 94305
DEK@Sail.Stanford.Edu

**Stephen C. Lipp**
University of Texas at Austin
Dept of Mechanical Engineering
ETC II, 4.152
Austin, TX 78712

**Pierre A. MacKay**
Northwest Computer Support Group
University of Washington
Mail Stop DW-10
Seattle, WA 98195
206-543-6259; 545-2386
MacKay@June.CS.Washington.edu

**Laurie Mann**
Stratus Computer
55 Fairbanks Boulevard
Marlboro, MA 01752
617-460-2610
uucp: harvard!anvil!es!Mann

**Charles-Michel Marle**
Dept. de Mathématiques
Université Pierre et Marie Curie
4, place Jussieu
75252 Paris Cedex 05, France
43.36.25.25

**Gérald Masini**
CRIN
B.P. 239
54506 Vandoeuvre-les-Nancy Cedex
France
+33 83.91.21.45

**Robert W. McGaffey**
Martin Marietta Energy Systems, Inc.
Building 9104-2
P. O. Box Y
Oak Ridge, TN 37831
615-574-0618
McGaffey%ORN.MFEnet@nmfecc.arpa

**Marie McPartland-Conn**
GTE Laboratories
40 Sylvan Rd
Waltham, MA 02254
617-466-4220

**David Ness**
TV Guide
Radnor, PA 19088
215-293-8860

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

**Mitch Pfeffer**
Suite 90
148 Harbor View South
Lawrence, NY 11559
516-239-4110

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

**Craig Platt**
Dept of Math & Astronomy
Machray Hall
Univ of Manitoba
Winnipeg R3T 2N2, Manitoba, Canada
204-474-9832
CSnet: platt@uofm.cc.cdn
Bitnet: platt@uofmcc

**Thomas J. Reid**
Computing Services Center
Texas A&M University
College Station, TX 77843
409-845-8459

**Barry Smith**
Kellerman & Smith
534 SW Third Ave
Portland, OR 97204
503-222-4234; TLX 9102404397
Usenet: tektronix!reed!barry

**Rilla Thedford**
Intergraph Corporation, MS HQ1006
One Madison Industrial Park
Huntsville, AL 35807
205-772-2440

**Graham Toal**
6 Pypers Wynd
Prestonpans
East Lothian, Scotland
United Kingdom
+44 31 661 8105
Bitnet: gtoal%uk.ac.ed@ukacrl

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

**Glenn L. Vanderburg**
Computing Services Center
Texas A & M University
College Station, TX 77843
409-845-8459
Bitnet: X230GV@TAMVM1

**Samuel B. Whidden**
American Mathematical Society
P. O. Box 6248
Providence, RI 02940
401-272-9500
sbw@Seed.AMS.com

**Ken Yap**
Dept of Computer Science
University of Rochester
Rochester, NY 14627
Ken@cs.Rochester.edu
Usenet: ..!{allegra,decvax,seismo,
cmcl2,harvard,topaz}!rochester!ken

**Hermann Zapf**
Seitersweg 35
D-6100 Darmstadt
Federal Republic Germany

# General Delivery

## From the President

Bart Childs

I started my last column with:

> The last issue of TUGboat (Vol. 8, No. 1) represents a measure of success in my mind. The number of contributions, their content, and all other measures of quality made it interesting and useful. I hesitate to mention any one paper, but a number of people have commented about their high interest on several of the papers. Let's keep up the good work.

I am sure that my words were not the reason that TUGboat Vol. 8, No. 2 was the best yet. It was a dramatic improvement again in the most important area, your contributions. In those two issues we are already ahead of our total number of pages from any previous year. It really gave me a high to go to meetings and be able to show it.

Now, the same is true about the annual meeting which was held at the University of Washington. Dean Guenther did an outstanding job of pulling the program together, with assistance from Pierre MacKay. The University of Washington and particularly Donna Gardner were outstanding hosts. The meeting had record attendance, the courses were large, the weather was wonderful, and it was an all-around good event.

For good news and bad news, Donald Knuth did not attend (that is bad news) but it was a successful meeting anyway. Maybe we can help realize Don's goal of TEX's being viable without his involvement. He will always be the Grand Wizard.

Thanks to Patrick Ion, Mike Ferguson, and Cal Jackson for being the nominating committee. Patrick prefaced his introduction of nominees for President and Treasurer with the comment that they "expected to find that TUG members are very busy people." I was re-elected President, which is rewarding. That is taken as a compliment, but the long range quality of our organization is dependent on having a large and representative number of us participating as authors, contributors, volunteers, and officers. I can't imagine conditions being such that I would accept again in two years.

David Ness has been elected Treasurer. We welcome him to the Steering and Finance Committees. Since TUG is now a much bigger business, his willingness to work and proximity to Rhode Island mean that we will have a good replacement for Sam Whidden's yeoman service.

The Vice President has been assigned a new duty: coordination of volunteers. The list of potential volunteers compiled from the questionnaires of two years ago is now in Rilla Thedford's hands; if you showed interest then in becoming a volunteer, Rilla will be getting in touch with you. If you have an idea for something you can do that will benefit other TUG members, get in touch with Rilla.

Several items were discussed that we need to move forward with, as a society. The most pressing three are:

1. distribution standards for TEX systems,
2. printer (driver) standards, and
3. expanded member services, particularly communications and local user groups.

I have furnished a short article on the first item (page 263), Robert McGaffey's article in the last issue (page 161) and his committee trying for a written standard addresses the second, and we are working on the third. Ray Goucher is collecting samples of organizational criteria to be distributed to the Steering Committee for action.

I just received the following from Hope Hamilton of NCAR. It appeared in the `Staff Notes` for NCAR, Vol. 22, No. 39.

> **TEX Users' Group Meeting**
>
> The first Boulder-area TEX Users' Group meeting was held in the Fleishmann Building on 15 September. About 30 users attended from the University of Colorado, the National Oceanic and Atmospheric Administration, Ball Aerospace, and NCAR. Sheridan Garcia (Climate and Global Dynamics Division), who organized the meeting and served as moderator, provided TEX software handouts and cautioned the participants on several dangerous turns along the way to perfect electronic typesetting. Participants agreed that at the October meeting, a comparison between TEX and its macro-packaged spinoff, LATEX, should be featured. Meetings will be held monthly; for more information, call Sheridan on ext. 1358.

Should we have a section of TUGboat dedicated to such reports? We would appreciate receiving such information and any by-laws that local organizations of this type have established.

Happy TEXing.

## The Volunteer Tree

Rilla Thedford

Congratulations to all who attended the Seattle Users Group meeting. It was great to see so many new faces and lots of the veteran faces as well. (We sure missed you, Don.)

As y'all could tell there is an ever growing amount of activity in the TEX community. In order to channel this activity and enthusiasm, it has been suggested that various committees or special interest groups be created. It has also been suggested that the Vice President of TUG take on that responsibility. Well, here I am, but where are you!!! We need your help!

We should start with the major areas of concern and "branch" out from there, with several interest areas under each major area. The major areas are:

- Program coordination
- Standards
  - TUG representative to X3VI standards committee
  - Printer standards – hardware/software
  - Distribution tapes – (already lots of discussion in this area)
  - New versions of TEX and METAFONT
  - Manuals of documentation standards
  - Macro writing and documentation
- Fonts
  - Font catalog
  - Font exchange
  - Naming conventions
- Publications
  - TUGboat
  - TEXniques – new manuscripts
  - Software catalog
  - Index of TUGboat articles
- TEXhax monitor
- Macros and Packages library
- "Masters" Coordinator
  - Aid in production of TUGboat
  - Compile list of members willing to take technical calls
- Employment bulletin board
  - Classified ads in TUGboat
  - Job openings/jobs wanted

Several people have already volunteered and many more have suggested special interest areas. Below is the current list of interests, and the name(s) of volunteers, if known. If you have any interest in participating, *please please* contact me or a volunteer already listed.

Group: Beginning Users
Name(s): Open

Group: METAFONT
Name(s): Doug Henderson (UC-Berkeley)

Group: Implementors (as opposed to users)
Name(s): Open

Group: Macros (standards, collecting/collating packages)
Name(s): Bill Kaster (Personal TEX)

Group: Commercial applications
Name(s): Elizabeth Barnhart (TVGuide)

Group: Law applications
Name(s): Allen Dyer

Group: LATEX
Name(s): Jackie Damrau (Univ. of New Mexico)

Group: Speakers forum
Name(s): Gary Benson (Los Alamos)

Group: Regional/State Chapters
Name(s): Robert Messer

Group: Primers
Name(s): Michael Doob (Univ. of Manitoba, Winnipeg)

Group: Printer standards, QA files
Name(s): Robert McGaffey (Martin Marietta, Oak Ridge, Tenn.)

Group: Index of old TUGboats
Name(s): James Slagle (TVGuide)
(There is another person who expressed an interest; I apologize for misplacing your name.)

Group: Do's & Dont's
Name(s): Open

Group: Linguistics and the Humanities
Name(s): Christina Thiele (Carleton Univ., Ottawa)

Group: Basic TEX questions
Name(s): Doris Hsia (Stanford)

The above people have already expressed an interest in special areas. Let's take these and group them under the major areas. This is just the "tip of the tree" and a great start.

If I have forgotten your name, please forgive and drop me a line reminding me of your special interest. If you want to become involved in the "Volunteer Tree" please contact me.

Intergraph
Rilla Thedford - MS: HQ1006
1 Madison Industrial Park
Huntsville, AL 35806
1-205-772-2440

## Book Publishing using TEX

Charles-Michel Marle
Université Pierre et Marie Curie

I was very interested by the paper by Tony Siegman, "Book publishing using TEX", in the April 1987 issue of TUGboat. I think that my experience on this subject, more limited, but somewhat different, may be of interest for some readers of TUGboat.

A few years ago I wrote, with a coauthor, Paulette Libermann, a book in French, on symplectic geometry. A Dutch publisher, D. Reidel Publishing Company, offered to us to pay a translator and to publish the book in English. But he asked us to provide a "camera ready" text. At that time, I knew that an implementation of TEX existed in France at the University of Strasbourg, but I had no knowledge at all about its use. However, I thought that it would be much more interesting for me to use TEX than to type the translation of my book on an ordinary typewriter for producing the camera ready sheets. The translator, Bertram Schwarzbach, was willing to use a microcomputer for typing the translation. So I thought that once the translation was made, I would have only to introduce the mathematical formulas and the formatting TEX commands.

I obtained from Dominique Foata a preprocessor, developed at the University of Strasbourg and called STRATEC, made for Victor Sirius microcomputers, which makes much easier the typing of mathematics, and also of French texts. The keyboard, which is specially redefined, allows the typing of all the Greek letters and of the most usual mathematics symbols (such as $\in$, $\sum$, $\mapsto$, $\int$, $\cap$, ...), with a single keystroke, or maybe with the `ALT` key or the `SHIFT` key and a single keystroke. It allows also the typing of all the French accented letters é, è ê, à, ù, ...) in exactly the same way as with a French typewriter. Moreover, all these special symbols, when typed, appear on the screen with their usual shape, not as TEX control sequences. Once the text is registered in a file, the preprocessor STRATEC produces a standard TEX file by mapping all the special symbols onto the corresponding TEX control sequences. Moreover it checks a few points of TEX syntax, such as the balance between the number of opening and closing braces, the consistency of the mode changes (between ordinary text and mathematics) and of the font changes (for the most usual fonts, roman, italics and boldface). This preprocessor, which was used to produce the book, was certainly a great help (for typing the mathematics, not for the French accented letters, since the book is in English). It is a pity that, to my knowledge, there is no equivalent product at a reasonable price for microcomputers more common than the Victor Sirius, which is no longer manufactured.

Dominique Foata gave me also a small macro package which was developed at the University of Strasbourg for producing a special issue of the French mathematics journal *Astérisque* devoted to "Élie Cartan et les mathématiques d'aujourd'hui". With his help, and the help of Jacques Désarménien, I made a few changes and additions to adjust this macro package to my needs.

When the work began, I had no implementation of TEX in my university. Therefore, during more than one year, most of the text was typed without any control, except an output of a few pages from the first chapter, made by Dominique Foata at the University of Strasbourg, from a diskette sent by mail. Then my university bought a PCTEX implementation of TEX, and I was able to check my typing. An output of the whole book, except the index, the list of references and the table of contents, was produced on a dot matrix printer at the beginning of July 1986. At that time, there was no laser printer available at my university. I spent two months correcting my text, making the index, the list of references and the table of contents, checking and inserting in the text the numbers corresponding to the papers in the list of references. Then I made another output on a dot matrix printer for checking the final line breaks and page breaks, and for determining the final page numbers to be introduced in the table of contents and the index.

The final output was made on a laser printer at the computer center of the CNRS in Strasbourg, with the help of Georges Weil. A curious thing happened there, which seems to me worth reporting here. I took to Strasbourg, on IBM PC diskettes, the source files (`*.TEX` files) and the `*.DVI` files (one `*.DVI` file for each chapter) produced by PCTEX. We transferred these files to the mainframe computer on which TEX and the printer driver were installed. For most of the chapters, we made the printing directly from the `*.DVI` files, without running TEX again. But for some unknown reason, the transfer of the `*.DVI` file of chapters 2 and 3 failed. So we ran TEX on the corresponding `*.TEX` files to produce locally the `*.DVI` files for these two chapters. The output of chapter 2 was what I expected. But that of chapter 3 was not: a page break occurred a little earlier than in the output made previously in Paris on a dot matrix printer, the following page breaks were slightly shifted a few

lines up, and finally, the output of chapter 3 had one page more than what I expected. This was very annoying, since the page number of the last page of chapter 3 was then the same as that of the first page of chapter 4, and some of the page numbers listed in the index were now wrong. I do not know why two different TEX implementations, one on a microcomputer, and another on a mainframe computer, applied to the same `*.TEX` source files, can give page breaks at different places. Maybe it is due to differences in the accuracy of the arithmetic in the two computers. Finally, the problem was solved by trying again and again to transfer the `*.DVI` files obtained with the IBM PC, until we succeeded.

In conclusion, I may say that I am very happy to have learnt the use of TEX while writing a book, and that I have the deepest respect for Donald Knuth, the author of this very remarkable mathematical typesetting system. Now I use it also for my mail, my research papers and the papers I give to my students. If I ever write another book, I will certainly use it again, but I will try to use more of its advanced features (for automatically making the index, checking cross references, and so on). I will try also to obtain from the publisher an output on a phototypesetter, since my book, produced from camera ready sheets made with a laser printer, although better than most books produced from typewritten camera ready sheets, is not quite as beautiful as true typeset books.

## TEX Training, etc. — A TUG Meeting Trip Report

Laurie Mann
Stratus Computer

The TEX Users Group Meeting was very informative. It was held at the University of Washington in Seattle this year. There were about 160 attendees from all over North America, Europe, and Japan. There was one track of programming, with some discussion groups on Wednesday. I attended the conference with my manager (Carol Klos) and stayed on campus. The view of the Cascade Mountains and Lake Washington from the odd-numbered rooms of McMahon Hall was spectacular.

The conference opened unofficially on Sunday night, with a "get-acquainted" cruise on Lake Washington. The weather was wonderful and we all got a good look at the "Lake Houses of the Washington Rich and Famous". Carol and I spent most of the evening with Jackie Damrau (last year's Knuth Scholarship winner) and Marie McPartland-Conn (one of this year's Knuth Scholarship winners).

The official opening was Monday morning, with a talk from Bart Childs on where TUG is and where it's going. The group now has 2,500 individual members, and about 140 institutional members. The user base of TEX has expanded dramatically over the last few years, with the introduction of some TEX-based software packages. TEX is shifting from being almost exclusively a university-based tool to being used in many corporations for publications. A newspaper in Kansas is being produced with TEX. **TV GUIDE** is experimenting to see if TEX can produce 190 different editions of the magazine each week.

The theme of this year's conference was "TEX and the Humanities". The first speaker, Christina Thiele, discussed how she uses TEX to produce a scholarly linguistics journal (*Canadian Journal of Linguistics*) several times a year, as well as an annual conference proceedings. She described some of the difficulties in typesetting for linguistics publications. She also stressed the importance of keeping the editing and production cycles separate. One interesting trend Christina mentioned is the proliferation of PC use by writers for the journal. In 1984, only 11% of the writers used PCs. By 1986, this had risen to 68%.

A number of speakers described the challenges behind developing fonts and TEX for non-Roman alphabets. Silvio Levy talked about how he developed a more readable modern Greek font. He also traced the history of Greek typography. Walter Andrews and Pierre MacKay presented a talk on publishing in Turkish. Nobuo Saito and Yasuki Saito spoke on how jTEX has gotten started in Japan. Since Japanese is an ideographic rather than an alphabetic language, and uses thousands of ideographs rather than a handful of letters, implementing a set of Japanese characters in TEX has been difficult. While sites using alphabetic languages can load information for many different fonts, only 10 fonts can be loaded for Japanese. jTEX now has 50 users and is considered stable.

A big issue raised during the week was that of device drivers. In the last *TUGboat*, Robert McGaffey called for standardization of device drivers. Standardization is important for two reasons:

1. The more output devices you have, the harder it is to maintain the driver software.
2. People are selling drivers. Since drivers are being sold, standardization protects the consumer.

The standardization issue is very complicated, due to the wide variety of output devices and their assorted capabilities. Several program items dealt with drivers in different ways. Robert led a panel discussion on the value of driver standardization, and whether or not it could even be done. The next day, Nelson Beebe presented information on how he develops drivers and what he thought constituted a good driver. He believed drivers should incorporate the following features:

1. Uniform UNIX-like command line interface on all machines for all output devices.
2. Control of page selection and order from the command line.
3. Default and user-specified font substitutions.
4. Default directory paths overridden by environment variables and command line options.
5. User control of magnification.
6. Virtual font caching.
7. Multiple output copies.
8. Multiple input files.
9. Debugging output, including open file tracing.
10. Page origin positioning.
11. Autospool (TOPS-20 only).
12. Supporting any combination and order of font formats.

There seemed to be more vendors this year. Personal TEX, ArborText, FTL systems, Kellerman & Smith, and Addison-Wesley were among those attending. There are now several variants of TEX for the Mac and the IBM PC. Some software houses are taking a different approach to TEX, by implementing a system of menus to supply the end-user interface to TEX. K-Talk Communications is testing a product that would take a file formatted with WordPerfect and generate a TEX source file. TEXnology, Inc. presented a macro package called MacroTEX which features a number of interesting table-building macros.

TUG made a PC available for software swapping; this was a very helpful innovation. Karen and Bill Butler spent a good part of Tuesday and Wednesday copying software from disk to disk. This was an excellent vehicle for sharing macros, style files, documentation, drivers, font files, etc., and I hope more people use it next year.

One very interesting talk about TEX was give by David Ness of **TV GUIDE**. For the last three years, **TV GUIDE** has been experimenting with TEX, to see if it can be used to produce the weekly magazine. One major reason **TV GUIDE** is experimenting with TEX is because they believe TEX will weather foreseeable changes in typesetting technology. Currently, **TV GUIDE** is using a "home grown" database and typesetting system for production. Now, the staff uses TEX for most internal documentation, and continues to test it for magazine production.

Part of Tuesday afternoon was spent in "Birds-of-a-Feather" sessions — small discussion groups where people can discuss TEX-related issues. The groups this year discussed driver standardization, TEX and graphics files, TEX training, editing and production cycles, and non-English TEX. I was interested in three of the topics, and chose the training group. Stephan v. Bechtolsheim led about thirty of us in a discussion on how to make TEX training more manageable. The problems in developing TEX training programs and aids include the following:

1. Insufficient support from university or corporate management for TEX training.
2. New TEX users must learn some programming.
3. Distinctions between non-technical and technical TEX users need to be drawn.

Many sites have been developing in-house training and introductory TEX manuals. There was some interest in TUG collecting introductory materials so that they could be made available to people who wanted to use them. (It isn't clear to me that TUG has the time or resources to do this, but it would be terrific if they could.)

The meeting ended with some discussion on how to train the trainer. Ways to improve this process include the publication of a "Train the Trainer" manual, and a "Train the Trainer" course.

Christina organized a Wednesday breakfast discussion group on how to document macros. About twenty of us met at 8 a.m. to discuss some strategies. We agreed that the following features should be part of a standard macro description:

1. Macro definition
2. Description of the macro parameters and the global variables
3. When the macro is used
4. How it's used
5. Typeset sample of the macro output
6. Related macros
7. How to change the macro definition
8. Some of the error messages this macro may elicit

9. Origin of the macro (particularly helpful when macros are shipped off to other TEX sites).

The meeting wrapped up on Wednesday, with presentations made by Bart Childs to members of the TEX community who had made special contributions. Each received an appropriately inscribed plaque, and Barbara Beeton received an original Duane Bibby illustration. Wednesday also featured a talk by Alan Dyer on how the Maryland Bar Association has made TEX a voluntary standard for legal typesetting in Maryland. Barbara led the annual "TEX Question and Answer" session, and that was the "official" end of the meeting. Unofficially, some people stayed on for hours.

On Thursday and Friday, I attended Stephan Bechtolsheim's output routines course with about 15 other people. This was the first formal TEX course I'd ever taken, and I certainly hope to take another one in the future. Stephan's class notes are extremely helpful.

While most of the conference was spent attending meetings and the like, there were opportunities to meet TEX users from all over. Addison-Wesley sponsored a barbecue on Monday, and Personal TEX put on a wine and cheese party on Tuesday. Seattle has many ethnic restaurants, and a number of seafood restaurants on renovated piers near the downtown area.

My only regret from the week I spent in Seattle was that I was unable to take a trip to the mountains. Marie McPartland-Conn, who made the trip out to Mt. Saint Helens, described the destruction of the mountainside and the tons of dust still found in the area. She had the dirty car to prove the point!

My sight-seeing was pretty limited to watching the mountains from the back of the dorm, and a few trips to downtown Seattle. One night, I joined a group that went to Tillicum, a recreation of an Indian village on Burke Island in Seattle harbor. This spot featured an Indian-style salmon dinner, Indian dancing, and Indian crafts. Friday night, Marie, Doris Hsia (the other Knuth Scholarship winner), Michael Doob and I had one last ethnic meal and our last good look at downtown Seattle from the top of the Space Needle.

In my opinion, Dean Guenther, of Washington State University, Donna Gardner, of the University of Washington, Ray Goucher and Bart Childs all did an excellent job in pulling together the conference and the courses.

So, Ray, when do we get to go to Seattle again????????

### From the Editor

Barbara Beeton

First of all, I'd like to thank the TUG hierarchy for the most delightful token of appreciation presented to me at the Seattle meeting. Those who weren't present can see it in a different context on page 56 of *The TEXbook*. The original drawing has been beautifully matted and framed, and was given to me accompanied by a brass plaque purporting to explain why I deserve it. I shall treasure them both.

I'd also like to report on some activity outside of TUG relating to an item that appeared in TUGboat. Chuck Bigelow's article, "Notes on typeface protection" (Vol. 7, No. 3), has now been reprinted in both English and French (translated by Jacques André, a "charter" TUG member, with the title "Du piratage de fontes" in *Technique et Science Informatiques*). I replied to a recent request to reprint this article yet again with an offer to provide the text on floppy disk. The response to my offer seems to me a backhanded comment on what I think TEX and TUG are all about:

> "I have asked our Production Department whether a copy of the article on IBM PC diskette would be useful, but they think it is preferable to go ahead with the traditional type-setting. Thank you, in any case, for the offer."

Under the circumstances, I shall let the source remain nameless.

## Software

### Proposed Minimum Standards for TEX Distributions

Bart Childs

We have an activity beginning for definition of a standard for drivers. A standard for TEX distributions will probably be harder in some sense because it has so many parts. The purpose of this is to create a start of a series of discussions that might lead to someone volunteering to lead such an effort.

I offer some thoughts here that I think are appropriate for inclusion (at least in part) for a minimum TEX distribution standard. Many of the thoughts are not original with me because I have

observed some of the other distributions for micros through mainframes. I also learned a lot from recent opportunities to watch someone else install TEX from my instructions.

I will not try to name directories here because they should vary greatly based on the restrictions of the operating system and other arbitrary restrictions. However, I will offer some suggestions on naming conventions of files that contain pixels.

### Files That Contain Pixels

These files should be named in a convention that is not misleading! Some distributions use `dpi329` to indicate files for 300 pixels/inch (dpi, from dots/inch) at `\magstephalf`. I propose that: *One directory should contain all the pixel files for a given resolution and marking technology.* Thus, a directory of `pk_b300` would contain files for a **write black 300 dpi** engine. Extensions should be added for aspect ratios. The Computer Modern roman 10pt font should exist with several magnifications. Extensions need to reflect the magnification and storage protocol. I propose using `0pk`, `hpk`, ... `7pk` for the obvious zero, half, ... seven magnifications. The storage protocols should be `pk`, `px`, and `gf`. Are there others? The three-character extensions will enable use of these files on mainframes and minicomputers that act as servers for PC interconnect systems which are appearing. These files must obey MS-DOS restrictions.

### TEX Source Distributions

TEX distributions on magnetic media that will hold 25Mbyte and above should always contain the sources that are on the standard distribution. Diskette distributions should have sources available in some form. Since many micros don't have sufficient storage for the sources, they should probably be kept separate and charged for separately. (Personal TEX has this available now.) I am not sure that we can expect the (proprietary?) change files to be included, but I would like that.

### Formats

The standard formats should be included: plain TEX, LATEX, SLITEX, and AMS-TEX. Further, these should include locals like `\today`, `time` (giving `\thetime` and `\miltime`), or a facility for including them in a standard local.

We should expect users to create local or personal macros. We should also expect installations to begin using METAFONT for creating specialized fonts and logos. Thus, each distribution should include the means for rebuilding these formats.

### Maintenance

Each distribution should have an explicit option for maintenance and provisions of updates. The regularity of these updates will be dependent on extras that might be furnished.

### Evolution

The creation of standards for drivers will certainly cause some standard macros to be furnished for TEX that will specify how graphics is to be included. These items, along with the coming of color, duplex print engines, and other improvements in technology, indicate that many changes will be made.

I suggest the following handling for *invisible* fonts, as in SLITEX: *all fonts whose names begin with "i" shall be considered invisible. The procedure that creates the file name can remove the "i" from the actual file name and open the visible file thereby getting the correct dimensions.* Drivers can follow the same strategy and carry an extra variable for each font. If this variable indicates that the font is *invisible*, then the driver has to be changed only in the `set` and `put` routines. They would simply move rather than output inkless fonts.

### Utilities

Each distribution should have a utility that converts pixel files from the other two(?) standard formats to the preferred distribution formats. Thus, our DG distribution should have `GFtoPK` and `PXtoPK`.

We should develop and distribute conversion packages for older formatting systems and common word processors into TEX and/or LATEX.

I think that we should have language specific (such as German, French, etc.) translators into plain ASCII. The ß and accents like those on page 135 of the TEXbook are frequently entered as one keystroke. Some editors have the capability of converting it to the TEX control sequences, but the utility would be handy too. (Is this part of Michael Ferguson's multilingual TEX?)

Finally, we should have utilities to aid in detecting brace mismatches, removing TEX commands and checking for spelling, and easier public table macros. Some of these exist, but are not in `WEB`.

### Summary

I hope these issues start some discussion on what we should have in TEX distributions. I have not touched on other items like user interfaces, editor macros, and font substitution which I think are a little further into the future. Do we have any volunteers to study distributions and try to work on a standards document?

## TANGLE Modification Causes Problems in METAFONT and PK Files

Thomas J. Reid
Texas A&M University

FOREWORD: Notice to sites using a version of TANGLE which preserves case for Pascal identifier names. This includes some implementations under VAX/VMS and most under AOS/VS. Sites running TANGLE under other operating systems may also be affected.

In tracking down a problem recently encountered in some PK files created by GFtoPK Version 1.3 under VAX/VMS it was found that TANGLE was generating incorrect Pascal source code. The problem shows up in GFtoPK in section 75 ⟨Write two-byte short character preamble⟩. The WEB statement:

```
flag_byte := flag_byte +
             comp_size div 65536 + 4 ;
```

was converted by TANGLE into:

```
flag_byte:=flag_byte+
        comp_size div 65540;
```

TANGLE has provisions which prevent it from adding constants when one is adjacent to a "*," "/," "MOD" or "DIV." However, the VAX/VMS change file which is used here (and at many other sites as well) modifies TANGLE to preserve the case of Pascal keywords and identifiers; the keywords "div" and "mod" were not being recognized as "DIV" and "MOD," respectively, because of the case difference.

The TANGLE bug is easily fixed by revising sections 105 ⟨Contribution is * or / or DIV or MOD⟩ and 110 ⟨If previous output was DIV or MOD⟩ to include tests for "div" and "mod." The problem can be fixed by inserting the WEB code at the end of this article into your change file for TANGLE. Again, this fix only applies to versions of TANGLE which have been modified to keep Pascal identifiers in lowercase.

### Implications of Bug

TANGLE was run before and after making the fix to determine the extent of its impact. The programs tested consist of

| | | | |
|---|---|---|---|
| WEAVE | GFtoPK | PKtype | POOLtype |
| TANGLE | GFtoDVI | PXtoPK | TFtoPL |
| BIBTEX | MFT | PKtoPX | PLtoTF |
| METAFONT | GFtoPXL | TEX | DVItype |
| GFtype | | | |

Of these, only METAFONT, GFtoPK and PXtoPK were affected.

GFtoPK was wrong only in section 75 ⟨Write two-byte short character preamble⟩ while PXtoPK was affected in sections 44 (**Writing the packed file**) and 69 ⟨Write two-byte short character preamble⟩. The GFtoPK and PXtoPK bugs result in unusable PK files for large size fonts (cminch, some magsteps of cmr17, etc.). METAFONT was affected in five statements: one statement in section 232 (**procedure** *init_big_node*) and four statements in section 566 (**procedure** *wTek_coor*). Section 566 was modified to support on-line displays on a Tektronix terminal, so the invalid code in this section does not impact any fonts created by METAFONT. In section 232, the statement

```
name_type(q+s):=half(s)+x_part_sector;
```

is converted by TANGLE into the remarkable

```
mem[q+s].hh.b1:=(s)div 7;
```

through the wonders of WEB macros. The correct Pascal code for the statement should be

```
mem[q+s].hh.b1:=(s)div 2+5;
```

The impact of this bug has not yet been assessed, although it does not seem to have made very much difference on any of our fonts.

### Additions to WEB change file for TANGLE

```
@x
@ @<Contribution is \.*...@>=
((t=ident)and(v=3)and@|
 (((out_contrib[1]="D")and(out_contrib[2]="I")and(out_contrib[3]="V")) or@|
 ((out_contrib[1]="M")and(out_contrib[2]="O")and(out_contrib[3]="D")) ))or@|
@^uppercase@>
 ((t=misc)and((v="*")or(v="/")))
@y
@ @<Contribution is \.*...@>=
((t=ident)and(v=3)and@|
 (((out_contrib[1]="D")and(out_contrib[2]="I")and(out_contrib[3]="V")) or@|
 ((out_contrib[1]="d")and(out_contrib[2]="i")and(out_contrib[3]="v")) or@|
 ((out_contrib[1]="M")and(out_contrib[2]="O")and(out_contrib[3]="D")) or@|
 ((out_contrib[1]="m")and(out_contrib[2]="o")and(out_contrib[3]="d")) ))or@|
@^uppercase@>
 ((t=misc)and((v="*")or(v="/")))
@z

@x
@ @<If previous output was \.{DIV}...@>=
if (out_ptr=break_ptr+3)or
 ((out_ptr=break_ptr+4)and(out_buf[break_ptr]=" ")) then
@^uppercase@>
  if ((out_buf[out_ptr-3]="D")and(out_buf[out_ptr-2]="I")and
    (out_buf[out_ptr-1]="V"))or @/
     ((out_buf[out_ptr-3]="M")and(out_buf[out_ptr-2]="O")and
    (out_buf[out_ptr-1]="D")) then@/ goto bad_case
@y
@ @<If previous output was \.{DIV}...@>=
if (out_ptr=break_ptr+3)or
 ((out_ptr=break_ptr+4)and(out_buf[break_ptr]=" ")) then
@^uppercase@>
  if ((out_buf[out_ptr-3]="D")and(out_buf[out_ptr-2]="I")and
    (out_buf[out_ptr-1]="V"))or @/
     ((out_buf[out_ptr-3]="d")and(out_buf[out_ptr-2]="i")and
    (out_buf[out_ptr-1]="v"))or @/
     ((out_buf[out_ptr-3]="M")and(out_buf[out_ptr-2]="O")and
    (out_buf[out_ptr-1]="D"))or @/
     ((out_buf[out_ptr-3]="m")and(out_buf[out_ptr-2]="o")and
    (out_buf[out_ptr-1]="d")) then@/ goto bad_case
@z
```

## Hyphenation Exception Log

Below is a list of words that TEX fails to hyphenate properly. This is the annual update; the list last appeared in Volume 7, No. 3, on page 145. Everything listed there is repeated here.

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; note that all forms must be specified in a `\hyphenation{...}` list if they occur in your document.

Thanks to all who submitted entries to the list. Several of the suggestions demonstrated a lack of familiarity with some of the rules of the hyphenation algorithm, so here is a short reminder of the relevant idiosyncrasies. Hyphens will not be inserted before the second letter, nor after the third-from-last letter of a word; thus no word shorter than five letters will be hyphenated. (For the details, see *The TEXbook*, page 454.) This particular rule is violated in some of the words listed; however, if a word is hyphenated correctly by TEX except for "missing" hyphens at the beginning or end, it has not been included here.

Some other permissible hyphens have been omitted for reasons of style or clarity. While this is at least partly a matter of personal taste, an author should think of the reader when deciding whether or not to permit just one more break-point in some obscure or confusing word. There really are times when a bit of rewriting is preferable.

One other warning: Some words can be more than one part of speech, depending on context, and have different hyphenations; for example, 'analyses' can be either a verb or a plural noun. If such a word appears in this list, hyphens are shown only for the portions of the word that would be hyphenated the same regardless of usage. These words are marked with a '*'; additional hyphenation points, if needed in your document, should be inserted with discretionary hyphens.

The reference used to check these hyphenations is *Webster's Third New International Dictionary*, Unabridged.

| | |
|---|---|
| `academy` | acad-e-my |
| `al-ge-brais-che` | al-ge-brai-sche |
| `anal-yse` | an-a-lyse |
| `anal-y-ses` | analy-ses * |
| `anomaly(ies)` | anom-aly(ies) |
| `anti-nomy(ies)` | an-tin-o-my(ies) |
| `an-tirev-o-lu-tion-ary` | an-ti-rev-o-lu-tion-ary |
| `ap-pendix` | ap-pen-dix |
| `asymp-totic` | as-ymp-tot-ic |
| `at-mo-sphere` | at-mos-phere |
| `ban-dleader` | band-leader |
| `Be-di-enung` | Be-die-nung |
| `be-haviour` | be-hav-iour |
| `bib-li-ographis-che` | bib-li-o-gra-phi-sche |
| `bid-if-fer-en-tial` | bi-dif-fer-en-tial |
| `biomath-e-mat-ics` | bio-math-e-mat-ics |
| `bornolog-i-cal` | bor-no-log-i-cal |
| `Brow-n-ian` | Brown-ian |
| `buz-zword` | buzz-word |
| `cartwheel` | cart-wheel |
| `cholesteric` | cho-les-teric |
| `columbia` | co-lum-bia |
| `congress` | con-gress |
| `Czechoslo-vakia` | Czecho-slo-va-kia |
| `database` | data-base |
| `dat-a-p-ath` | data-path |
| `defini-tive` | de-fin-i-tive |
| `democratism` | de-moc-ra-tism |
| `de-mos` | demos |
| `dis-tribute` | dis-trib-ute |
| `Di-jk-stra` | Dijk-stra |
| `duopolist` | du-op-o-list |
| `duopoly` | du-op-oly |
| `dy-namis-che` | dy-na-mi-sche |
| `eco-nomics` | eco-nom-ics |
| `economist` | econ-o-mist |
| `elec-trome-chan-i-cal` | electro-mechan-i-cal |
| `elec-tromechanoa-cous-tic` | electro-mechano-acoustic |
| `En-glish` | Eng-lish |
| `equiv-ari-ant` | equi-vari-ant |
| `Eu-le-rian` | Euler-ian |
| `ex-traor-di-nary` | ex-tra-or-di-nary |
| `Febru-ary` | Feb-ru-ary |
| `fermions` | fermi-ons |
| `flowchart` | flow-chart |
| `Forschungsin-sti-tut` | For-schungs-in-sti-tut |
| `funk-t-sional` | funk-tsional |
| `Gaus-sian` | Gauss-ian |
| `ge-o-met-ric` | geo-met-ric |
| `gnomon` | gno-mon |
| `Greif-swald` | Greifs-wald |
| `Grothendieck` | Grothen-dieck |
| `Grundlehren` | Grund-leh-ren |
| `Hamil-to-nian` | Hamil-ton-ian |
| `heroes` | he-roes |
| `Her-mi-tian` | Her-mit-ian |
| `hex-adec-i-mal` | hexa-dec-i-mal |
| `holon-omy` | ho-lo-no-my |
| `ho-mo-th-etic` | ho-mo-thetic |
| `id-iosyn-crasy` | idio-syn-crasy |
| `ig-nores-paces` | ignore-spaces |
| `in-finitely` | in-fin-ite-ly |

| | |
|---|---|
| in-finites-i-mal | in-fin-i-tes-i-mal |
| in-fras-truc-ture | in-fra-struc-ture |
| in-ter-dis-ci-plinary | in-ter-dis-ci-pli-nary |
| Japanese | Japan-ese |
| jeremi-ads | je-re-mi-ads |
| Kadomt-sev | Kad-om-tsev |
| Ko-rteweg | Kor-te-weg |
| Leg-en-dre | Le-gendre |
| Le-ices-ter | Leices-ter |
| Lip-s-chitz(ian) | Lip-schitz(-ian) |
| macroe-co-nomics | macro-eco-nomics |
| Manch-ester | Man-ches-ter |
| manuscript | man-u-script |
| marginal | mar-gin-al |
| Marko-vian | Mar-kov-ian |
| Mas-sachusetts | Mass-a-chu-setts |
| met-a-lan-guage | meta-lan-guage |
| mi-croe-co-nomics | micro-eco-nomics |
| mi-crofiche | mi-cro-fiche |
| mis-ogamy | mi-sog-a-my |
| mod-elling | mod-el-ling |
| molecule | mol-e-cule |
| mo-noen-er-getic | mono-en-er-getic |
| monopole | mono-pole |
| monopoly | mo-nop-oly |
| monos-pline | mono-spline |
| monos-trofic | mono-strofic |
| mul-ti-pli-ca-ble | mul-ti-plic-able |
| mul-tiuser | multi-user (better with explicit hyphen) |
| ne-ofields | neo-fields |
| Noethe-rian | Noe-ther-ian |
| none-mer-gency | non-emer-gency |
| nonequiv-ari-ance | non-equi-vari-ance |
| noneu-clidean | non-euclid-ean |
| non-i-so-mor-phic | non-iso-mor-phic |
| nonpseu-do-com-pact | non-pseudo-com-pact |
| non-s-mooth | non-smooth |
| No-ord-wi-jk-er-hout | Noord-wijker-hout |
| oligopolist | oli-gop-o-list |
| oligopoly | oli-gop-oly |
| paradigm | par-a-digm |
| parabolic | par-a-bol-ic |
| parametrized | pa-ram-e-trized |
| paramil-i-tary | para-mil-i-tary |
| petroleum | pe-tro-le-um |
| phe-nomenon | phe-nom-e-non |
| Poincare | Poin-care |
| polyene | poly-ene |
| poly-go-niza-tion | polyg-on-i-za-tion |
| poroe-las-tic | poro-elas-tic |
| postam-ble | post-am-ble |
| Po-ten-tial-gle-ichung | Po-ten-tial-glei-chung |
| pream-ble | pre-am-ble |
| preloaded | pre-loaded |
| pre-pro-ces-sor | pre-proces-sor |
| pro-cess | process |
| 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 |
| quasir-ad-i-cal | qua-si-rad-i-cal |
| quasiresid-ual | qua-si-resid-ual |
| qua-sis-mooth | qua-si-smooth |
| qua-sis-ta-tion-ary | qua-si-sta-tion-ary |
| qu-a-si-tri-an-gu-lar | qua-si-tri-an-gu-lar |
| re-ar-range-ment | re-arrange-ment |
| Rie-man-nian | Rie-mann-ian |
| righ-teous(ness) | right-eous(-ness) |
| schedul-ing | sched-ul-ing |
| schot-tis-che | schot-tische |
| Schrodinger | Schro-ding-er |
| Schwarzschild | Schwarz-schild |
| semidef-i-nite | semi-def-i-nite |
| semi-ho-mo-th-etic | semi-ho-mo-thet-ic |
| ser-vomech-a-nism | ser-vo-mech-anism |
| setup | set-up |
| severely | se-vere-ly |
| solenoid | so-le-noid |
| spheroid | spher-oid |
| stan-dalone | stand-alone |
| statis-tics | sta-tis-tics |
| stochas-tic | sto-chas-tic |
| Stokess-che | Stokes-sche |
| summable | sum-ma-ble |
| tele-g-ra-pher | te-leg-ra-pher |
| tech-nis-che | tech-ni-sche |
| ther-moe-las-tic | ther-mo-elas-tic |
| times-tamp | time-stamp |
| ve-r-all-ge-mein-erte | ver-all-ge-mein-erte |
| Verteilun-gen | Ver-tei-lun-gen |
| vs-pace | vspace |
| Wahrschein-lichkeit-s-the-o-rie | Wahr-schein-lich-keits-the-o-rie |
| waveg-uide | wave-guide |
| whitesided | white-sided |
| whites-pace | white-space |
| widespread | wide-spread |
| Winch-ester | Win-ches-ter |
| workhorse | work-horse |
| wraparound | wrap-around |
| Yingy-ong Shuxue Jisuan | Ying-yong Shu-xue Ji-suan |

# Fonts

## Update: METAFONT mode_def Settings for Various TEX Output Devices

Doug Henderson

Hi,

I am introducing myself in this Update column since I will be trying to keep track of the METAFONT **mode_def** parameters for the TEX Users Group. My name is Doug Henderson, I work at the Division of Library Automation at U.C. Berkeley, and I am this year's METAFONT coordinator. If you have comments or questions about creating **mode_def**'s for new printers or about using existing settings for printers you can send them to me at either my bitnet address of `dlatex@cmsa.berkeley.edu` or my physical address in Berkeley:

Doug Henderson
186 University Hall
Berkeley, CA 94720

Please give me a while to respond, as we all know that many TEX Users Group people are volunteer workers who have jobs that must take priority, and if I take longer than a week, feel free to harass me again.

It seems that there is enough concern out there regarding METAFONT fonts and how they relate to printers (and typesetters) that we will maintain an ongoing list keeping track of established **mode_def** settings for these and new printers. If there are significantly many (more than one) contributions I will submit the **mode_def** chart to be printed in the next TUGboat. Below is a clone of the chart settings last submitted by Barbara Beeton with a few new printer contributions I have added.

The table on the next page contains a summary of the relevant settings gleaned from available sources. Most of the print engines cited in the table are listed below, along with an indication of whether they are write-black (wb) or write-white (ww), if known, and the names of some of the output devices into which they have been built.

| | |
|---|---|
| Canon CX (wb) | Apple LaserWriter; Cordata; HP LaserJet, LaserJet 500 Plus; Imagen 8/300; QMS and Talaris 8 ppm printers |
| Canon CXD (wb) | Imagen 3308 ImageServer XP |
| Canon SX (??) | HP LaserJet Plus, LaserJet Series II; Imagen ImageStation/S, 2308/S |
| Canon LBP-10 | Imagen 10/240 |
| Canon (wb) | Imagen 3320, Imagen 7320 |
| Ricoh 4081 (ww) | DEC LN03; TI OmniLaser 2115 |
| Ricoh LP4120 (ww) | HP 2688A, Imagen 12/300 |
| Ricoh 4150 (ww) | Talaris 1500 |
| Xerox XP-12 (ww) | DEC LN01, QMS 1200, Talaris 1200, Xerox 2700 |
| Xerox XP-24 (ww) | Imagen 24/300, QMS 2400, Talaris 2400, Xerox 3700 |

As always, additions and corrections to this list are solicited.

A note from John Lavagnino of Brandeis University warns against assuming that "improved" models of printers, or even printers from different manufacturers based on the same print engine, will produce equivalent output:

> We have discovered that the LN03 and the LN03-Plus don't print the same way: a font that looks fine on the LN03 will look lighter on the LN03-Plus. In fact it isn't necessary to download fonts to observe this: even the internal fonts look different.
>
> We've been badgering DEC about this for some time, and they have finally agreed that this is the case. The current story is that they "made the pixels smaller" on the LN03-Plus, "to make it look more like a typewriter."

Recently, there has been a similar discovery with regard to the HP LaserJet vs. HP LaserJet Plus printer. Ian Young writes

> I was wondering whether the "wimpiness" of the characters I am seeing on the printer I am using (HP LaserJet Plus) could be possibly due to my having generated the fonts originally for a 'real' LaserJet, which I believe has a different print engine.

I called the folks at Hewlett-Packard and Canon, and between the two found out that the engine for the HP LaserJet was, as we knew, the Canon LBP-CX. The surprising thing was that they used a different print engine, the Canon LBP-SX for their new printer the HP LaserJet Plus. So Ian, I guess it's right you are, you seemed to have used fonts built for the CX and new parameters are needed for the newer SX style. It is unclear whether this engine is write-white or not. Seems sort of strange that neither HP nor Canon knew whether it was a write-white engine or not. Oh well. Please keep us posted on your results. Anyone else do this yet?

Again, please be aware that a good **mode_def** for one will only be a poor approximation for the other.

Consider yourselves warned.

Typical **mode_def** parameter settings for CM fonts

| Source of information | | *pixels_per_inch* | *blacker* | *fillin* | *o_correction* | *aspect_ratio* |
|---|---|---|---|---|---|---|
| PLAIN.MF | | | | | | |
| proof | | 2601.72 | 0 | 0 | 1 | |
| lowres | | 200 | .65 | .2 | .4 | |
| WAITS.MF | | | | | | |
| dover | (Xerox Dover) | 384 | 1.2 | 0 | .6 | |
| imagen | (Canon CX) | 300 | 0 | .2 | .6 | |
| qms | (Xerox XP-12E) | 300 | .75* | 0* | .5* | |
| aps | (APS-Micro5) | 722.909 | .2 | .2 | 1 | |
| crs | (Alphatype CRS) | 4000+4000/3 | .4 | 0 | 1 | |
| boise | (HP 2680A) | 180 | .55 | .1 | .3 | |
| DD | (DataDisc terminal) | 70 | 0 | 0 | .2 | |
| canon | (Canon LBP-10) | 240 | .2 | .2 | .4 | |
| newDD | (DataDisc terminal) | 70 | 0 | 0 | .2 | 4/3 |
| cg | (Compugraphic 8600) | 1301.5 | .2 | .2 | 1 | 1569/1301.5 |
| epson | | 240 | 0 | 0 | .2 | 9/10 |
| Charles Karney, TEXhax 86#4, Oct 86 [Note 1] | | | | | | |
| qms | (Xerox XP-12E) | 300 | .8 | .2 | .4 | |
| John Gourlay, May 87 [Note 2; TUGboat **8**#2, page 10] | | | | | | |
| xeroxxxvii | (Xerox XP-12) | 300 | .6 | −.3 | .6 | |
| Charles LaBrec, TEXhax 86#6, Oct 86 [Note 3] | | | | | | |
| decln | (Ricoh 4080) | 300 | .9 | −.2 | .5 | |
| Stan Osborne, Apr 87 [Note 4] | | | | | | |
| decln | (Ricoh 4080) | 300 | .2 | −.4 | .5 | |
| Janene Winter, May 87 [Note 5; TUGboat **8**#2, page 20] | | | | | | |
| ibm | (IBM 3820) (wb) | 240 | .65 | −.2 | .45 | |
| ibm-a | (IBM 3812) (wb) | 240 | .4 | −.2 | .4 | |
| ibm-b | (IBM 3800) (wb) | 240 | .2 | −.1 | .6 | |
| ibm-c | (IBM 4250) (wb) | 600 | .05 | 0 | .6 | |
| sherpa | (IBM 6670) (ww) | 240 | 1 | 1 | .6 | |
| Matthias Feyerabend, GSI, Darmstadt, May 87 [Note 6] | | | | | | |
| ibmlaser | (IBM 3820) | 240 | .3 | −1 | .4 | |
| bensmall | (Benson 9211) | 200 | −.5 | 0 | .4 | |
| bensmall | [alternate settings for problem fonts] | 200 | 0 | 0 | .4 | |
| benbig | (Benson 9436) | 254 | −.8 | 0 | .4 | |
| benbig | [alternate settings for problem fonts] | 254 | −.1 | 0 | .4 | |
| Doug Henderson, DLA, Berkeley, Sep. 87 [Note 7] | | | | | | |
| varityper | (Varityper VT-600) | 600 | .2 | .2 | .8 | |
| linoone | (Linotronic 100) | 1270 | .2 | .2 | 1 | |
| linothree | (Linotronic 300) | 2540 | .2 | .2 | 1 | |

* A note in WAITS.MF states that these settings are conjectural.

Typical **mode_def** parameter settings for CM fonts

| Source of information | | *pixels_per_inch* | *blacker* | *fillin* | *o_correction* | *aspect_ratio* |
|---|---|---|---|---|---|---|
| PLAIN.MF | | | | | | |
| proof | | 2601.72 | 0 | 0 | 1 | |
| lowres | | 200 | .65 | .2 | .4 | |
| WAITS.MF | | | | | | |
| dover | (Xerox Dover) | 384 | 1.2 | 0 | .6 | |
| imagen | (Canon CX) | 300 | 0 | .2 | .6 | |
| qms | (Xerox XP-12E) | 300 | .75* | 0* | .5* | |
| aps | (APS-Micro5) | 722.909 | .2 | .2 | 1 | |
| crs | (Alphatype CRS) | 4000+4000/3 | .4 | 0 | 1 | |
| boise | (HP 2680A) | 180 | .55 | .1 | .3 | |
| DD | (DataDisc terminal) | 70 | 0 | 0 | .2 | |
| canon | (Canon LBP-10) | 240 | .2 | .2 | .4 | |
| newDD | (DataDisc terminal) | 70 | 0 | 0 | .2 | 4/3 |
| cg | (Compugraphic 8600) | 1301.5 | .2 | .2 | 1 | 1569/1301.5 |
| epson | | 240 | 0 | 0 | .2 | 9/10 |
| Charles Karney, TEXhax 86#4, Oct 86 [Note 1] | | | | | | |
| qms | (Xerox XP-12E) | 300 | .8 | .2 | .4 | |
| John Gourlay, May 87 [Note 2; TUGboat **8**#2, page 10] | | | | | | |
| xeroxxxvii | (Xerox XP-12) | 300 | .6 | −.3 | .6 | |
| Charles LaBrec, TEXhax 86#6, Oct 86 [Note 3] | | | | | | |
| decln | (Ricoh 4080) | 300 | .9 | −.2 | .5 | |
| Stan Osborne, Apr 87 [Note 4] | | | | | | |
| decln | (Ricoh 4080) | 300 | .2 | −.4 | .5 | |
| Janene Winter, May 87 [Note 5; TUGboat **8**#2, page 20] | | | | | | |
| ibm | (IBM 3820) (wb) | 240 | .65 | −.2 | .45 | |
| ibm-a | (IBM 3812) (wb) | 240 | .4 | −.2 | .4 | |
| ibm-b | (IBM 3800) (wb) | 240 | .2 | −.1 | .6 | |
| ibm-c | (IBM 4250) (wb) | 600 | .05 | 0 | .6 | |
| sherpa | (IBM 6670) (ww) | 240 | 1 | 1 | .6 | |
| Matthias Feyerabend, GSI, Darmstadt, May 87 [Note 6] | | | | | | |
| ibmlaser | (IBM 3820) | 240 | .3 | −1 | .4 | |
| bensmall | (Benson 9211) | 200 | −.5 | 0 | .4 | |
| bensmall | [alternate settings for problem fonts] | 200 | 0 | 0 | .4 | |
| benbig | (Benson 9436) | 254 | −.8 | 0 | .4 | |
| benbig | [alternate settings for problem fonts] | 254 | −.1 | 0 | .4 | |
| Doug Henderson, DLA, Berkeley, Sep. 87 [Note 7] | | | | | | |
| varityper | (Varityper VT-600) | 600 | .2 | .2 | .8 | |
| linoone | (Linotronic 100) | 1270 | .2 | .2 | 1 | |
| linothree | (Linotronic 300) | 2540 | .2 | .2 | 1 | |

* A note in WAITS.MF states that these settings are conjectural.

**Notes:**

1. Charles Karney states, "...I haven't fully explored the parameter space. If anyone knows of a better (or 'authorized') solution, I'd appreciate hearing about it."
   [Karney%PPC.MFEnet@LLL-MFE.Arpa]
2. John Gourlay has diagnosed an unexpected modification to the pen path as *blacker* increases, causing the diameter of such letters as "o" to decrease; the details are discussed in his article in TUGboat **8**#2, page 20. The parameter values given here are a compromise, allowing most characters to keep their original sizes, although the value of *blacker* "is not quite enough to compensate for the thinning inherent in the printer." There is still "an inconsistency in the weights of characters. Nevertheless, [Gourlay] feel[s] that this set of parameters is considerably better than the ones that result from the 'conjectural' parameters, and also better than the 'am' fonts they replace."
   Gourlay.Ohio-State@csnet-relay
3. Charles LaBrec's comments: "I have twiddled the parameters a bit, and this seems to produce good 12 point cm fonts. I am a bit unsure because changing *blacker*, *fillin*, or *o_correction* seem to make no difference for quite a large range of values. I can't remember exactly, but you will get the same results as [these] for $.4 < blacker < .9$, $-.8 < fillin < -.1$, and $0 < o\_correction < .7$. But this probably makes a good starting point."
   [crl@newton.physics.purdue.edu]
   [Editor's note: The value given in TUGboat 8#1 for `decln` *fillin* should have been $-.2$, not $+.2$.]
4. Stan Osborne: "The `decln` mode [Mr. LaBrec] suggested did not *fillin* correctly and was too black for the smaller point sizes. His choice of settings produces small sized fonts that are much blacker than the small cmr's found in the cmr book (Vol E). ... I found the [above] values of *blacker* and *fillin* to produce readable small fonts for an LN03.... These values were not carefully tested for larger point sizes. (I stopped experimenting when I got something I liked and I had verified that larger sizes were also usable.) [...!ucbvax!dual!dbi!stan]
5. Janene Winter has found these settings "to be optimal for the IBM printers". This information was transmitted by Dean Guenther along with his site report (TUGboat **8**#2, page 10).
6. Matthias Feyerabend: "Fonts tested are CMR5, CMR10, CMR12 and CMSSI17 for a full range of settings for *blacker* and *fillin*."
7. Doug Henderson: Preliminary reasonable looking fonts produced for these three printers. Since the Linotype typesetters are fairly expensive I cannot do extensive testing. Anyone have one and want to donate some use for testing? Please let me know.
   [dlatex@cmsa.berkeley.edu]

## Halftone Output from TEX

Adrian F. Clark

Don Knuth's article in TUGboat volume 8 number 2 described the development of a number of fonts which allow halftone output—pictures—to be incorporated into TEX documents. This article chronicles the author's experiments into halftone production on a particular computer/laser printer combination, VAX/VMS and the LN03. It is important to understand that the picture is actually *typeset*, not just inserted into the final output by some printer-specific `\special` command; the following results can, in principle, be achieved on *any* output device using a perfectly normal implementation of TEX.

In the image processing field, where the author works, technical reports are invariably crammed with halftone output. The conventional method of reproducing pictures is photographically. This is slow and expensive, particularly for internal reports with small distributions. Moreover, unless great care is taken over the photographs—using a flat-screen CRT, calibrating films, standardising the processing, and so on—much of the visual impact can be lost. Hence, the possibility of incorporating imagery into TEX document without recourse to a dark room is very attractive.

A great deal of work has been carried out into the properties of the human eye. One result is that the eye is only really capable of distinguishing about 64 grey levels, although it is very good at detecting boundaries between regions of slightly differing grey level (see, for example, *"Digital Image Processing"* by R. C. Gonzalez and P. Wintz, published by Addison-Wesley in 1977). Another result is that the eye is much more sensitive to boundaries in dark regions than in light regions.

The halftone font used here is more or less the same as the 'double-dot' font described by Knuth. It has some 65 different grey levels, represented by the ASCII characters '0' (white) to 'p' (black). In principle, all one needs to do is to convert the grey levels of the individual pixels ("picture elements") of an image to the appropriate characters of the halftone font and sprinkle in a few TEX commands to ensure that the lines of the image are lined up in the output.

The only minor complication is that this sequence of characters includes '\', '^' and '_', which have special meanings to TEX. These must be treated specially. Knuth's approach was to delimit the picture data between macros, `\beginhalftone` and `\endhalftone`, which disable the special characters in a similar way to the 'verbatim' macros in Appendix E of *"The TEXbook"*. The approach developed by the author is much less elegant and builds larger disc files, but does not require special-purpose macros. Each line of the image is built up as a single `\hbox`. These lines are stacked into a `\vbox`, with the inter-line skip turned off. Finally, the `\vbox` is enclosed in another `\hbox`, which makes it easier to handle the picture in constructs such as `\centerline`. The scheme can be summarised as:

```
\hbox{ \vbox{ \halftone
   \offinterlineskip
   \hbox{...}
     ...
   \hbox{...}
}}
```

The `\halftone` command is used to select the halftone font, which must have been loaded with a command such as

```
\font\halftone=hf300
```

assuming the TFM file is called `HF300.TFM`.

A FORTRAN `SUBROUTINE`, `TEXPIC`, was written to output images to files in this format. The image is represented as a `REAL` array dimensioned as `(M,N)`, where `M` is the number of pixels per line and `N` the number of lines. (The use of a `REAL` array to hold data which are usually 8-bit may seem a little strange, but this representation has many advantages—for example, when Fourier transforming an image.)

Since we would normally like our pictures to have the best contrast, `TEXPIC` scans through the image to find its minimum and maximum, then scales the output to make full use of the grey levels in the halftone font. For most purposes, a single

```
CALL TEXPIC( PIC, M, N, FN )
```

is sufficient. `FN` is a `CHARACTER` variable or quoted string holding the output filename.

Of course, there are occasions when we would like to compare pictures, so fixing the contrast is sometimes desirable; hence, `TEXPIC` has associated routines to fix the range of intensities (`ZRANGE`) and re-select automatic intensity scaling (`ZAUTO`), which must be invoked before `TEXPIC` to have an effect. Similarly, `TEXPIC` can plot negative pictures as well as positive ones: `DONEG` tells it to output subsequent pictures as negatives and `DOPOS` returns it to the default state.

Inserting the picture into a document prepared with plain TEX is quite simple, using commands to generate a 'float', such as

```
\midinsert
   \centerline{\input picture}
\endinsert
```

for a picture in the file `PICTURE.TEX`. To draw a border around the picture, as for the examples presented here, one would define a macro `\border`

```
\def\border#1{\vbox{\hrule\hbox{
   \vrule\kern3pt\vbox{\kern3pt#1
   \kern3pt}\kern3pt\vrule}\hrule}}
```

The picture would then be set with

```
\centerline{\border{\input picture}}
```

The procedure with LATEX is somewhat different. The most sensible approach is to use the `figure` environment (*not* the `picture` environment)

```
\begin{figure}
   \centering
   \mbox{\input picture\relax}
   \caption{...}
\end{figure}
```

This generates a 'floating' figure, which usually surfaces at the top of the next page of output. The `\relax` following the filename in the `\mbox` command ensures that LATEX knows where the filename ends. To draw a border around the picture, replace the `\mbox` with a `\fbox`.

It is traditional to test out new image processing techniques on the 'girl' picture from the image database of the University of Southern California's Signal and Image Processing Institute. She is shown in Fig. 1 ($64 \times 64$ pixels). The output was plotted on a standard `LN03` laser printer using version 10 of Flavio Rose's `DVI2LN3`. For those unfamiliar with the `LN03`, it is a 300 dpi, white-writing laser printer

Figure 1: The Ubiquitous 'Girl' Image

based a Ricoh mechanism, supporting the downloading of fonts into on-board and plug-in RAM cartridges. The quality of Fig. 1 may not appear to be particularly good, but this is due to the comparatively low spatial resolution of the image data: approximately $256 \times 256$ pixels are needed to give a visually satisfying result—as we shall see.

Unfortunately, the standard LN03 will not output images of much greater than $64 \times 64$ pixels: if one tries to do so, it generates "band too complex" errors and produces broad white bands in the output. The actual cause of this is not known; however, it seems to be because the LN03 buffers plotting commands internally rather than writing their resulting glyphs into a bitmap. When the print operation actually starts, the driving microprocessor cannot translate the commands sufficiently quickly.

However, the LN03+ device (a field-installable hardware and firmware upgrade) has a full-page bitmap, and is quite capable of printing off large pictures. (However, a little care is needed in setting up the terminal line to which the printer is attached.)

There is another problem in producing these large pictures, and it concerns TEX itself. Since TEX was designed for typesetting text rather than pictures, its memory capacity is too small. Increasing the size of the memory (i.e., `mem_size`) is obviously feasible, at least on VAXen, but there is a snag: TEX was written to use 16-bit integers for subscripts into the memory arrays. However, the change file mechanism of WEB and the careful way in which TEX was written makes the conversion of 16-bit integers to 32-bit integers quite straightforward. (It is also necessary to disable some of TEX's initial consistency checking.)

When the author did this, producing a "big TEX", he found that the 16-bit and 32-bit versions of TEX were identical in almost every respect. The executable file was a few percent bigger, probably due to the increased memory space rather than the different integer representation. Likewise, the string pool and format files were slightly larger. However, there is *no* perceivable impact on execution times. (In fact, the author replaced the 16-bit version with big TEX without telling users—and no-one noticed any difference!)

This may seem a little surprising at first, but an examination of the (pseudo-) assembler generated by the PASCAL compiler provides the answer. The machine code generated for variables declared as `0..65535` (or, indeed, `0..255`) is *identical* to that for, say, `0..262144`; 32-bit integers are used in all cases. (This does, of course, not apply to **packed arrays**.) Moreover, TEX is very frugal in the way it handles its memory arrays, always re-using the same region if possible; this keeps the page fault rate low. Since the VAX initialises all memory to be 'demand-zero' when a program is loaded, there is no real increase in the system overhead due to unused regions of TEX's memory.

The version of TEX at the author's site has a large enough memory capacity for four $256 \times 256$ pictures (or one $512 \times 512$ picture!) in addition to the usual text, fonts and macro definitions. This allows users to put a few images into floating figures, as described above, without overflowing TEX's memory. For example, a $256 \times 256$ picture is shown in Fig. 2.

Indeed, to a certain extent, the physical size of a picture on the printed page determines the maximum number of pixels which can be plotted. Images of $512 \times 512$ pixels are more or less standard in the image processing community, while satellite images used in remote sensing applications have several thousand pixels on a side! Hence, if the image size exceeds a proscribed maximum (256 pixels, say), `TEXPIC` must *interpolate* between pixels to reduce the size of an image. Another associated `SUBROUTINE`, `TEXMAX`, is used to tell `TEXPIC` the maximum number of pixels which can be output. If the `M` dimension of an image exceeds this value, the image is interpolated down to this plottable maximum number of pixels.

There are many ways to perform the interpolation. The theoretical optimum is to use a $\sin x/x$ interpolation function (usually achieved via Fourier transformation), but this is slow. Cubic or linear interpolators tend to be used in practise. Recognising that TEX output of a reduced $4000 \times 4000$ pixel image will inevitably be inaccurate, `TEXPIC` uses a linear interpolation scheme. However, since linear interpolators usually blur edges (a particularly undesirable effect), it attempts to reduce the blur by using a *context-sensitive* interpolator. This in-

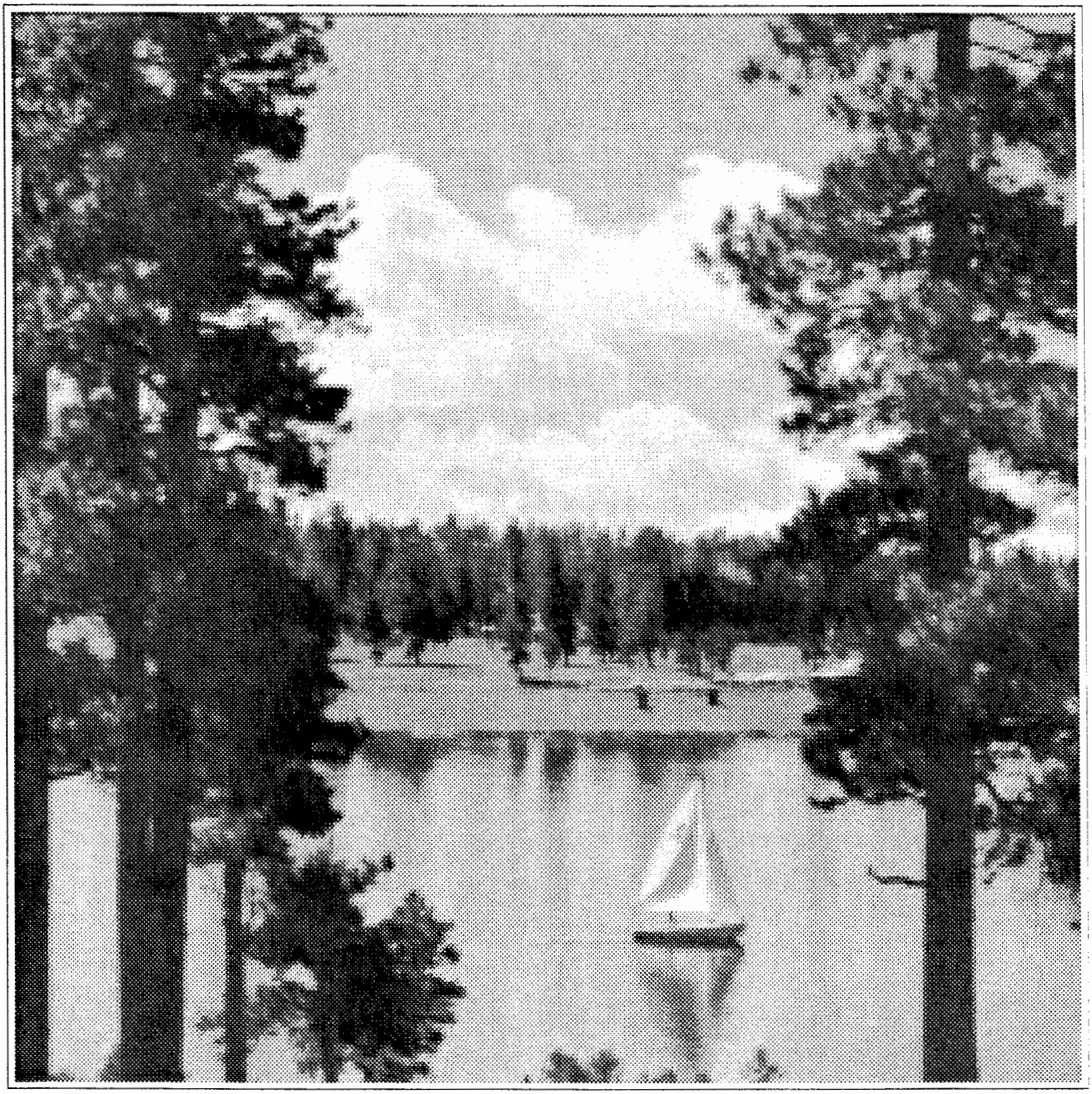


Figure 2: A 256 × 256 Lake Scene

terpolates between two sets of triplets of pixels at right angles and selects the value on the line with maximum gradient. For example, Fig. 3 is a 200 × 200 pixel image, reduced from a 512 × 512 image in this way.

All the software described here is available. TEXPIC and supporting routines exist in both standard FORTRAN and VAX FORTRAN; the VAX version does clever things with filenames and channel numbers. The big TEX change file incorporates an editor interface and automatic determination of batch or interactive use (see TUGboat vol. 8 no. 2 p. 177). It is, of course, specific to VMS, but may be useful for people making similar enhancements on other machines.

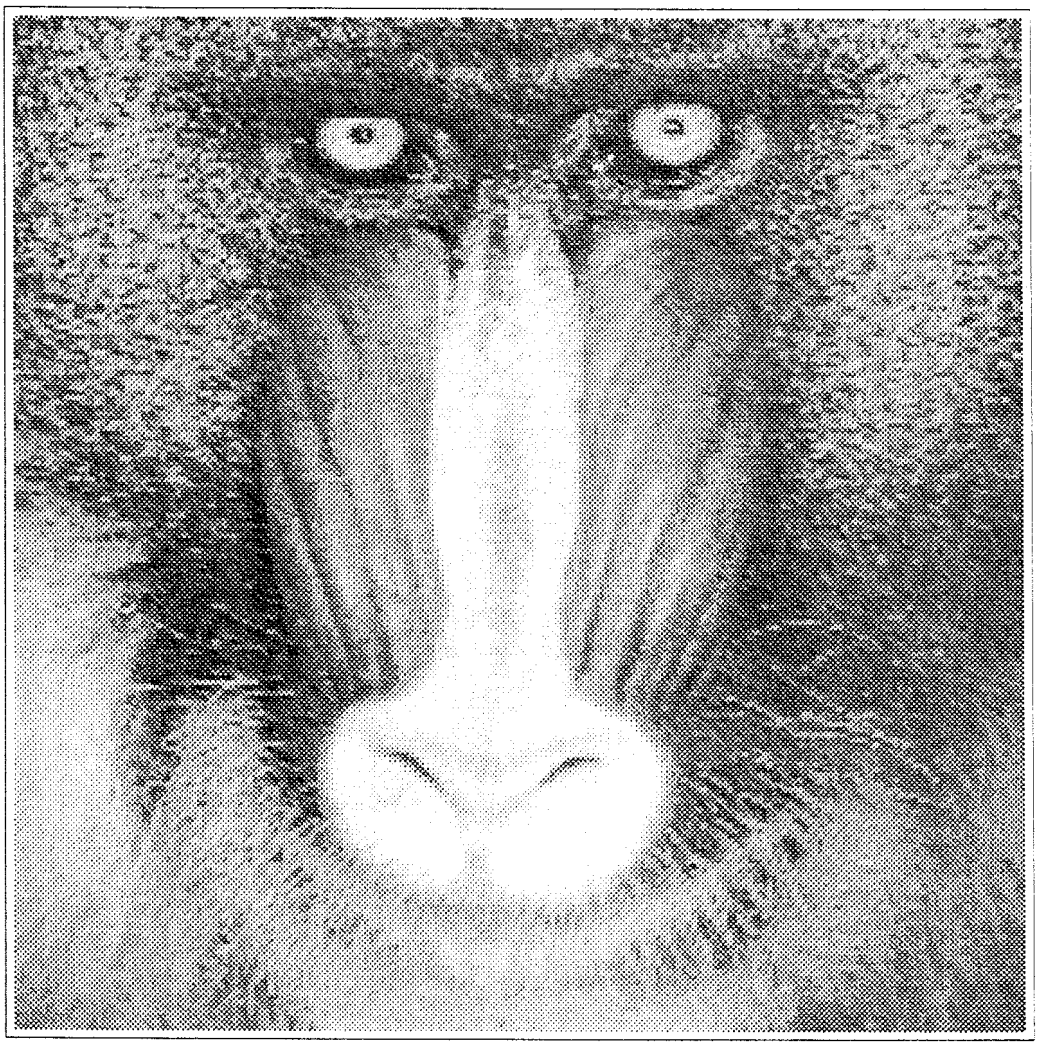


Figure 3: Mandrill Image, Reduced to 200 × 200Pixels from 512 × 512 Pixels

## Generating an *APL* Font

Aarno Hohti and Okko Kanerva
University of Helsinki

ABSTRACT. The *APL* language is well known for its peculiar symbols which have inhibited the use of this language in many programming environments. Making *APL* documents of good quality has been difficult and expensive. We describe here a simple way how to use METAFONT to generate an *APL* font for TEX by using existing font definitions as far as possible.

### Introduction

This note describes an interesting exercise in using METAFONT to produce new typefaces by combining letters from standard fonts. As we know, the *APL* language [6] of Kenneth Iverson has never gained the popularity it deserves largely because of its strange symbol set. Indeed, true *APL* users require a special keyboard to support the nonstandard but powerful operator symbols. Moreover, putting *APL* into print has always been a problem, and modern low cost computerized typesetting programs do not usually support *APL* style. TEX can be used to produce high quality printouts for technical text, and it would be desirable to have a possibility to mix in *APL* code. The companion program of TEX — METAFONT — provides a full means for a simple generation of an *APL* font for TEX, and the purpose of this note is to inform other people about the result we have obtained at the University of Helsinki. Let us note that there are at least three other *APL* fonts available for small computer environments. Indeed, the newsletter *APL Quote Quad** is produced by using Troff, and there exists a PostScript *APL* font for the Apple LaserWriter [5] and another font for the TEXT typesetting system [2].

Consider the following usual kind of function definition in *APL*:

```
      ∇S←SUMSQ N;I
[1]   S←I←0
[2]   →(N<I←I+1)/0
[3]   S←S+I*2
[4]   →2
[5]   ∇
```

How to write in this code, providing that we have a suitable font? The *APL* font should represent the screen output style of *APL* code and obey the same laws of spacing. Hence, it should be a typewriter-like typeface with fixed spacing; the same approach for representing TEX input was adopted by Knuth in the TEXbook. The *verbatim* macros have often been used for importing screen or paper outputs into TEX documents; some people misuse them for an easy construction of tables etc. In `verbatim`, the typewriter mode is entered by the control sequence `\begintt` — that mode is ended by `\endtt`. In the same vein, we could enter the *APL* mode by the control sequence `\beginapl`, and to end it by `\endapl`. However, it is more convenient to augment `verbatim` with `aplstyle` so that it can be used with several different typewriter-like fonts. (The *verbatim* macros can be found in the TEXbook, p. 421.) Since `@` (the *at sign*) is used as the escape character inside *verbatim* mode, our TEX code might (and in fact does) look as follows:

```
\choosett{apl}
\begintt
      @DL~S_SUMSQ N;I
 [1]  S_I_0
 [2]  @GO(N<I_I+1)/0
 [3]  S_S+I*2
 [4]  @GO~2
 [5]  @DL
\endtt
```

The control sequences `\DL` and `\GO` are not chosen arbitrarily but follow the conventions used in Digital's VAX *APL* interpreter [1]. As terminals usually do not support the *APL* character set, an alternative representation by two-letter mnemonics is provided by the interpreter. For the most part we have adopted these mnemonics also for our *APL* font. Hence, as an additional bonus the user should find it easy to combine his or her *APL* code with usual TEX code. Thus, it is not necessary for a VAX *APL* user to retype the definitions of *APL* functions in order to be able to use them in documents. However, the syntax of TEX code for *APL* text is somewhat different from that of Digital *APL*. Indeed, mnemonics must be followed by a non-letter (a character whose *catcode* is different from 12) and the escape character is `@`, not period as in the interpreter. Moreover, all spaces are obeyed; thus, if the user does not want a space after an *APL* symbol obtained from a control sequence, the *tilde* character (here a character with zero width) must be used as an end character. Using a macro for removing these differences would make `aplstyle` both slower and unnecessarily complicated.

* Newsletter of SIGAPL, the Special Interest Group for *APL*. Quote Quad has the same status in the world of *APL* as TUGboat in TEXnical world.

### The *APL* font table

*APL* symbols are divided into two classes: the primitive symbols and those obtained by overstriking two primitive ones. The overstrikes are traditionally obtained by typing the first symbol, by using backspace to go back one space and then typing the second symbol **over** the first one. However, in modern *APL* keyboards these double symbols are assigned to non-alphanumeric keys (for example, to keys under the *ALT* key). We decided to include only the primitive *APL* symbols in the font table; this enabled us to include also the lower-case letters, following modern conventions. (The original *APL* letters were restricted to capitals.) The comment symbol ⍝ is the only exception since it is keyed in as a double quote. Our font is a fixed size typeface with strongly slanted letters. Moreover, we have followed the style of best books in *APL*: all symbols should be drawn with a thin pen to get a touch of a typewriter. (This point is clearly witnessed, for example, in [5].) The places of some symbols are determined by the TEX font tables. For example, the hash sign # is used for the multiplication sign in VAX *APL*, and hence the corresponding symbol has the the same octal code (043) as the hash sign in TEX. The font table has the following form:

| | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
|---|---|---|---|---|---|---|---|---|
| *'000* | | ⍞ | ∇ | ∆ | ⍺ | ⍵ | ∈ | ¨ |
| *'010* | ∨ | ◊ | ≤ | ≥ | ⍴ | ⍳ | ∘ | ○ |
| *'020* | ∪ | ∩ | ⊂ | ⊃ | _ | ¯ | ~ | ≠ |
| *'030* | ⊤ | ⊥ | ⊣ | ⊢ | ⌊ | ⌈ | → | ↓ |
| *'040* | | ! | ⍝ | × | $ | ÷ | ∧ | ' |
| *'050* | ( | ) | * | + | , | - | . | / |
| *'060* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| *'070* | 8 | 9 | : | ; | < | = | > | ? |
| *'100* | | *A* | *B* | *C* | *D* | *E* | *F* | *G* |
| *'110* | *H* | *I* | *J* | *K* | *L* | *M* | *N* | *O* |
| *'120* | *P* | *Q* | *R* | *S* | *T* | *U* | *V* | *W* |
| *'130* | *X* | *Y* | *Z* | [ | \ | ] | ↑ | ← |
| *'140* | | *a* | *b* | *c* | *d* | *e* | *f* | *g* |
| *'150* | *h* | *i* | *j* | *k* | *l* | *m* | *n* | *o* |
| *'160* | *p* | *q* | *r* | *s* | *t* | *u* | *v* | *w* |
| *'170* | *x* | *y* | *z* | { | \| | } | | |

### The necessary METAFONT files

The whole process started when the first author had a paper containing *APL* symbols and was disappointed with the quality of the symbols available on the typewriter. Moreover, the secretary who had typed in the text had forgotten a couple of lines in the middle of the paper, and the correction of such mistakes seemed to be very clumsy in comparison with modern typesetting. Then he decided (together with the second author) to remedy the situation by creating an *APL* font for TEX he was using for other kinds of document. Many of the *APL* symbols needed were contained in standard fonts; for example, *diamond* can be found in *cmsy10*. For alphanumeric characters one could use *cmsltt10*. The simplest try for a solution of the problem would be to write a list of definitions that pick symbols from appropriate fonts. However, this brute force method does not really work since these symbols come from very different typefaces and, moreover, do not provide a fixed typeface. Hence, we decided to find an easy way of producing an *APL* font by using METAFONT.

As the starting point, we took the font *cmtex10*. This is a fixed typeface for an extended typewriter-like font including some Greek characters and mathematical symbols. The METAFONT file for this font, `cmtex10.mf` contains (as usual) a preamble that assigns values to several global variables, and the command `generate textset;`. Now the **driver** file `textset.mf` contains the commands `mode_setup;font_setup;` (establishing the values of the variables for this font) and several input files from which the METAFONT descriptions of the characters are to be found. Since some of these files treat the characters by name and since some of the definitions have to be changed (and some dropped), we considered it advisable to discard the driver file and to collect the separate METAFONT files, together with the preamble, to form a large single file `cmapl10.mf`. The `.mf` character files needed for `cmapl10.mf` are the following:

| | |
|---|---|
| `greekl` | (rho,omega,alpha) |
| `italms` | (iota) |
| `romand` | (roman digits) |
| `punct` | (punctuation symbols) |
| `romanp` | |
| `symbol` | (math symbols) |
| `sym` | |
| `romanu` | (upper case letters) |
| `romanl` | (lower case letters) |

### Definitions

First, put `font_identifier:="CMAPL";` and set `slant:=0;` in the preamble. Many definitions can be copied verbatim from the `.mf` files, but some of them need changes. The Greek *iota*, as given in `greekl.mf`, is strange to *APL* style; we use instead the *dotless i* from `italms.mf`. Moreover, we used the symbol *elt* (element) from `sym.mf` instead of the Greek epsilon. The symbols *del* and *delta* are

taken from *symbol.mf* (where their names are *large triangle* and *large inverted triangle*). However, they are too sturdy and too short to be placed in a proper *APL* font. Further, *del* must be lifted up so that it is vertically aligned with other symbols. The modifications are very easy to do, and the modified definition is shown below.

```
% sqrt48 was changed to 6.25 since the
% Del symbol in APL has a narrower top
% than the original reversed triangle
% symbol
% rule.nib has been changed to
%     light_rule.nib
% bot y3=-d-o has been changed to y3=0
% top y1=h-d has been changed to
%     top y1=h+2o

cmchar "Del";
beginchar(oct"002",16u#,asc_height#,0);
adjust_fit(0,0); pickup light_rule.nib;
top y1=h+2o; y2=y1; bot y3=0;
.5[x1,x2]=x3=good.x .5w; w:=r:=2x3;
lft x1=hround(.5w-u*6.25);
draw z1--z2--z3--cycle; % stroke
labels(1,2,3); endchar;
```

As can be seen from this example, the pen strokes were made thinner. Actually only one symbol was directly missing — this is the *quad box*. However, it can be obtained from the above by adding one control point:

```
cmchar "Quad";
beginchar(oct"001",16u#,body_height#,0);
adjust_fit(0,0); pickup light_rule.nib;
bot y1=0; y2=y1; top y3=h+2o; y4=y3;
.5[x1,x2]=x5=good.x .5w; w:=r:=2x5;
% The quad box is slightly wider than Del
lft x1=hround(.5w-u*7);
x3=x1; x4=x2;
draw z1--z2--z4--z3--cycle; % box
labels(1,2,3,4); endchar;
```

After taking care of the special symbols, the letters can be treated by finding a suitable value for *tilt ratio* (slant). Indeed, *APL* letters are **very** slanted. We suggest the value 1/5 for this font (the font *cmsl10* uses 1/6). One should remember to give the command *font_setup;* after setting *slant* to 1/5. Finally, one has to make the *verbatim* macros suitable for *APL* style. Since *verbatim* might be used for several different fonts in one document, we decided to include a control sequence *\choosett*.

```
\def\ifundefined#1{\expandafter
      \ifx\csname#1\endcsname\relax}
\outer\def\choosett#1{\ifundefined{#1}
      \message{Undefined font(?),
               replaced with cmtt10}
      \let\tt=\tentt
    \else
      \def\tt{\expandafter
              \csname#1\endcsname}\fi}
```

(The control sequence *\ifundefined* comes from the TEXbook, p. 308.) With the help of *\choosett*, the standard *verbatim* macro can be used without changes.

The *APL* symbols not in the font table are obtained — as usual — by overstriking two table symbols.

```
\newskip\charwidth
\def\overstrike#1#2{\setbox0=\hbox{#1}%
  \charwidth=\wd0 #1\hskip-\charwidth#2}
```

For example, the *grade up* and *grade down* symbols ∇ and ∆ are obtained by striking the *stile* symbol | over ∇ and ∆, respectively.

```
\def\GU{\overstrike{\DL}{\AB}}  % grade up
\def\GD{\overstrike{\LD}{\AB}}  % grade down
```

Now let us take another example of *APL*. Figure 1 shows some input and the resulting output.

### User extension

A modern user of a computerized typesetting facility will probably ask if it is possible to extend or modify fonts coming with the system. As with [5], where the font has an *analytic* and a *bitmapped* variant, we can distinguish between the need of modifying *cmapl10* via METAFONT and modification of the pixel files. *APL* symbols of various "blackness" or "thickness" may be desirable. Indeed, in [3] the user input is written with boldface *APL* symbols and the answers from the interpreter with thin ones. Furthermore, certain screen previewers use specific small size pixel files, and do not support the *APL* font.

The modification of the font by changing the values of some global variables in the preamble of the METAFONT file is easy and can be done by following how it is done in standard fonts such as *cmbx10* etc. This is the recommended way, too. However, if METAFONT is not available, then one must attack the pixel files. For direct hand editing, we use a program that converts a pixel file into a (bitmapped) text file acceptable to any standard screen editor, and another program reading the edited file back into a TEX pixel file. Scaling fonts

down to a desired size can be done in a similar (but automatic) manner; this facility is needed by a previewer (written by the first author) not using runtime scaling.

## References


[1] Digital Equipment Corporation: *VAX-11 APL Reference Manual*, 1983.

[2] Feldberg, Ian: *TEXT: Publication-Quality Characters Come To APL Graphics*, Proceedings of the 1986 *APL* Conference, SIGAPL, pp. 306 - 313.

[3] Gilman, Leonard, and Allen J. Rose: *APL, An Interactive Approach*, John Wiley & Sons, Inc., 1984.

[4] Grenander, Ulf: *Mathematical Experiments on the Computer*, Academic Press, 1982.

[5] Howland, John E.: *Typesetting APL using a Macintosh*, Proceedings of the 1986 *APL* Conference, SIGAPL, pp. 301 - 305.

[6] Iverson, Kenneth: *A Programming Language*, Wiley, New York, 1962.


## The address


University of Helsinki
Department of Mathematics
Hallituskatu 15
SF-00100 HELSINKI
FINLAND


**Figure 1. An *APL* example**

The input

```
\choosett{apl}
\setbox0=\vbox{\hsize=5.5truein
\begintt
    [0]   Z_A1 PROD A2;A;I;V
    [1]   "RETURNS THE PRODUCT OF THE POLYNOMIALS A1 AND A2
    [2]   "THE ARGUMENTS ARE GIVEN AS COEFFICIENT ARRAYS
    [3]   I_⍴⍴A1
    [4]   Z_A1⍴⍳0.+A2
    [5]   LOOP:V_⍴⍴Z ⍝DM V[(⍴⍴I)+1]-1
    [6]   V_⍴⍴Z_Z,[(⍴⍴I)+1]V⍴⍳0
    [7]   A_(1+-⍳⍴V[1])∘.⌊((⍳⍴⍴V)∈(1,(⍴⍴I)+1))/V)⍴1
    [8]   Z_+/[1]A⍴RV[(⍴⍴I)+1]Z
    [9]   ((⍴⍳⍴⍴Z)>⍴⍴I)/LOOP
\endtt
                }
$$\boxit{\boxit{\box0}}$$
\centerline{\sevenrm An APL function for polynomial multiplication}
```

gives the output

```
[0]   Z←A1 PROD A2;A;I;V
[1]   ⍝RETURNS THE PRODUCT OF THE POLYNOMIALS A1 AND A2
[2]   ⍝THE ARGUMENTS ARE GIVEN AS COEFFICIENT ARRAYS
[3]   I←⍴A1
[4]   Z←A1∘.×A2
[5]   LOOP:V←⍴Z ◇ V[(⍴I)+1]-1
[6]   V←⍴Z←Z,[(⍴I)+1]V⍴0
[7]   A←(1+-⍳V[1])∘.×((~(⍳⍴V)∈(1,(⍴I)+1))/V)⍴1
[8]   Z←+/[1]A⌽[(⍴I)+1]Z
[9]   ((⍴⍴Z)>⍴I)/LOOP
```

An APL function for polynomial multiplication

# Output Devices

## TEX Output Devices

Don Hosek

The device tables on the following pages list all the TEX device drivers currently known to TUG. Some of the drivers indicated in the tables are considered proprietary. Most are not on the standard distribution tapes; those drivers which are on the distribution tapes are indicated in the listing of sources below. To obtain information regarding an interface, if it is supposed to be included in a standard distribution, first try the appropriate site coordinator or distributor; otherwise request information directly from the sites listed.

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 full address and its source are shown. When information on the drivers is available, it is included below.

Screen previewers for multi-user computers are listed in the section entitled "Screen Previewers". If a source has been listed previously under "Sources", then a reference is made to that section for names of contacts, etc.

Corrections, updates, and new information for the list are welcome; send them to Don Hosek, Bitnet DHOSEK@HMCVAX (postal address, page 255).

### Sources

**ACC** Advanced Computer Communications, Diane Cast, 720 Santa Barbara Street, Santa Barbara, CA 93101, 805-963-9431 (DECUS, May '85)

**Adelaide** Adelaide University, Australia

The programs listed under Adelaide have been submitted to the standard distributions for the appropriate computers. The PostScript driver permits inclusion of PostScript files in a TEX file. The driver is described in *TUGboat, Vol. 8, No. 1.*

**AMS** American Mathematical Society, Barbara Beeton, 401-272-9500 Arpanet: BNB@Seed.AMS.com

**Arbor** ArborText, Inc., Bruce Baker, 313-996-3566, Arpanet: bwb%arbortext@umix.cc.umich.edu

ArborText's software is proprietary and ranges in price from $150 to $3000. The drivers for PostScript printers, the HP LaserJet Plus, the QMS Lasergrafix, and Imagen printers are part of their DVILASER series. The drivers all support graphics and include other special features such as use of resident fonts or landscape printing when supported by the individual printers.

Printing on the Autologic APS-5 and $\mu$-5 phototypesetters with DVIAPS includes support of Autologic standard library fonts and Logo processing.

**A-W** Addison-Wesley, Brian Skidmore, 617-944-3700, ext. 2253

Addison-Wesley supports graphics on all Macintosh software, and on Imagen, PostScript, and QMS laser printers on the IBM PC.

**Bochum** Ruhr Universität Bochum, Norbert Schwarz, 49 234 700-4014

**Caltech1** California Institute of Technology, Glen Gribble, 818-356-6988

**Caltech2** California Institute of Technology, Chuck Lane, Bitnet: CEL@CITHEX

**Canon** Canon Tokyo, Masaaki Nagashima, (03)758-2111

**Carleton** Carleton University, Neil Holtz, 613-231-7145

**CMU** Carnegie-Mellon University, Howard Gayle, 412-578-3042

**Columb.** Columbia University, Frank da Cruz, 212-280-5126

**COS** COS Information, Gilbert Gingras, 514-738-2191

**DEC** Digital Equipment Corporation, John Sauter, 603-881-2301

The LN03 driver is on the VAX/VMS distribution tape.

**GA Tech** GA Technologies

**GMD** Gesellschaft für Mathematik und Datenverarbeitung, Federal Republic of Germany, Dr. Wolfgang Appelt, uucp: seismo!unido!gmdzi!zi.gmd.dbp.de!appelt

**HMC** Harvey Mudd College, Don Hosek, Bitnet: DHOSEK@HMCVAX

**HP** Hewlett-Packard, Stuart Beatty, 303-226-3800

**IAM** Institut für Angewandte Math, Univ of Bonn, Federal Republic of Germany, Bernd Shulze, 0228-733427, Bitnet: BESCHU@DBNUAMA1

**INFN** INFN/CNAF, Bologna, Italy, Maria Luisa Luvisetto, 51-498286, BITnet: MILTEX@IBOINFN

The CNAF device drivers are on the VAX/VMS distribution tape.

**Intergraph** Intergraph, Mike Cunningham, 205-772-2000

**JDJW** JDJ Wordware, John D. Johnson, 415-965-3245, Arpanet: M.JOHN@Sierra.Stanford.Edu

**K&S** Kellerman and Smith, Barry Smith, 503-222-4234

The MacIntosh drivers and the VAX/VMS Imagen driver support graphics.

**LLL** Lawrence Livermore Laboratory

**LSU** Louisiana State University, Neal Stoltzfus, 504-388-1570

**Milan1** Università Degli Studi Milan, Italy, Dario Lucarella, 02/23.62.441

**Milan2** Università Degli Studi Milan, Italy, Giovanni Canzii, 02/23.52.93

**MIT** Massachusetts Institute of Technology, Chris Lindblad, MIT AI Laboratory, 617-253-8828

The drivers for Symbolics Lisp machines use the Symbolics Generic Hardcopy interface as a back end, so it should work on any printer that has a driver written for it. The printers listed in the table indicate drivers the program has been tested on.

The UNIX drivers for PostScript and QMS printers both support landscape printing and graphics inclusion via specials.

**MPAE** Max-Planck-Institut für Aeronomie, H. Kopka, (49) 556-41451, Bitnet: `MI040L@D6O6WD01`

**MR** Math Reviews, Patrick Ion, 313-996-5273

**NJIT** New Jersey Institute of Technology, Bill Cheswick, 201-596-2900, Arpanet: `cheswick@jvnca.csc.org`

**OCLC** OCLC, Tom Hickey, 6565 Frantz Road, Dublin, OH 43017, 616-764-6075

**OSU2** Ohio State University, John Gourlay, 614-422-1741, `gourlay.ohio-state@csnet-relay`

**Pers** Personal TEX, Inc., Lance Carnes, 415-388-8853

Graphic output is supported on Imagen, PostScript, and QMS printers.

**PPC** Princeton Plasma Physics Lab, Charles Karney, Arpanet: `Karney%PPC.MFENET@NMFECC.ARPA`

Versatec output from TEXspool is produced via the NETPLOT program. TEXspool also produces output for the FR80 camera. Color and graphics primitives are supported through specials.

**Procyon** Procyon Informatics, Dublin, Ireland, John Roden, 353-1-791323

**SARA** Stichting Acad Rechenzentrum Amsterdam, Han Noot, Stichting Math Centrum, Tweede Boerhaavestraat 49, 1091 AL Amsterdam (see *TUGboat, Vol. 5, No. 1*)

**Scan** Scan Laser, England, John Escott, +1 638 0536

**Sci Ap** Science Applications, San Diego, CA, 619-458-2616

**SLAC** Stanford Linear Accelerator Center, 415-854-3300

The SLAC drivers are on the standard CMS distribution tape.

**SRI** SRI International

**Stanford** Stanford University

The Imagen driver from Stanford is present on most distributions as the file `DVIIMP.WEB`. It provides limited graphics ability.

**Sun** Sun, Inc.

**Sydney** University of Sydney, Alec Dunn, (02) 692 2014, ACSnet: `alecd@facet.ee.su.oz`

**Talaris** Talaris, Rick Brown, 619-587-0787

All of the Talaris drivers support graphics.

**T A&M1** Texas A&M, Bart Childs, 409-845-5470, CSnet: `Childs@TAMU`

Graphics is supported on the Data General drivers for the Printronix, Toshiba, and Versatec on the Data General MV. On the TI PC, graphics is supported on the Printronix and Texas Instruments 855 printers. There are also previewers available for both the Data General and the TI.

**T A&M2** Texas A&M, Ken Marsh, 409-845-4940, Bitnet: `KMarsh@TAMNIL`

**T A&M3** Texas A&M, Norman Naugle, 409-845-3104

The QMS driver supports inclusion of QUIC graphics commands via specials as well as landscape printing.

**T A&M4** Texas A&M, Thomas Reid, 409-845-8459, Bitnet: `X066TR@TAMVM1`

The TEXrox package includes a DVI driver (TEXrox), a GF/PXL to Xerox font converter (PXLrox2), and a utility to build TFM files from licensed Xerox fonts (Xetrix). The programs are all written in C. Xetrix currently runs only under UNIX.

At present the TEXrox package is being distributed on a twelve-month trial basis; the trial is free for U.S. educational and government institutions, $100 for foreign or commercial institutions. Licensing agreements will be available when the trial offer expires.

**THD** Technische Hochschule Darmstadt, Klaus Guntermann, Bitnet: `XITIKGUN@DDATHD21`

Drivers developed at THD are not public domain; contact the distributors listed below for further information. All these drivers use PK format font files. Drivers for laser printers support graphics in device dependent format.

Drivers for the Kyocera F-1010 and F-2010 are distributed by LaserPrint, P. O. Box 35, D-6101 Fränkisch Crumbach, Federal Republic Germany; +49 6164 4044.

The preview driver (written in CWEB) has been ported to a VAXstation II GPX, using UIS library functions; for further information, contact Philips Kommunikations Industrie AG, TEKADE Fernmeldeanlagen, Attn. Dr. J. Lenzer, Thurn-und-Taxis-Str., D-8500 Nürnberg, Federal Republic Germany; +49 911 5262019.

Information on the VAX/VMS driver for Philips Elpho 20 can be obtained from Klaus Guntermann.

All other drivers, for the Atari ST, are distributed by Kettler EDV Consulting, P. O. Box 1345, D-8172 Lenggries, Federal Republic Germany; phone +49 8042 8081.

**Tools** Tools GmbH Bonn, Edgar Fuß, Kaiserstraße 48, 5300 Bonn, Federal Republic of Germany

The Tools implementation of TEX and the drivers listed are described in *TUGboat, Vol. 8, No. 1.*

**TRC Finland** Technical Research Centre of Finland, Tor Lillqvist, +358 0 4566132, Bitnet: `tml@fingate`

**UBC** University of British Columbia, Afton Cayford, 604-228-3045

**UCB** University of California, Berkeley, Michael Harrison, Arpanet: `vortex@berkeley.arpa`

**UCIrv1** University of California, Irvine, David Benjamin

**UCIrv2** University of California, Irvine, Tim Morgan, Arpanet: `Morgan@UCI`

**U Del** University of Delaware, Daniel Grim, 302-451-1990, Arpanet: `grim@huey.udel.edu`

The distribution includes a program to convert font files generated by METAFONT to Xerox font format.

**U Köln** Univ of Köln, Federal Republic of Germany, Jochen Roderburg, 0221-/478-5372, Bitnet: `A0045@DKORRZK0`

**U Mass** University of Massachusetts, Amherst, Gary Wallace, 413-545-4296

**U MD** University of Maryland, Chris Torek, 301-454-7690, Arpanet: `chris@mimsy.umd.edu`

The UNIX Imagen driver is on the UNIX distribution tape.

**U Mich** University of Michigan, Kari Gluski, 313-763-6069

**UNI.C** Aarhus University, Regional Computer Center

**U Shef** University of Sheffield, England, Ewart North, (0742)-78555, ext. 4307

**Utah** University of Utah, Nelson H. F. Beebe, 801-581-5254, Arpanet: `Beebe@Utah-Science`

The Beebe family of drivers was described in *TUGboat, Vol. 8, No. 1.* Graphics is supported only in the DVIALW (PostScript) driver.

**U Wash1** University of Washington, Pierre MacKay, 206-543-6259, Arpanet: `MacKay@June.CS.Washington.edu`

The programs listed under U Wash1 are all on the standard UNIX distribution tape.

**U Wash2** University of Washington, Jim Fox, 206-543-4320, Bitnet: `fox7632@uwacdc`

The QMS driver for the CDC Cyber was written under NOS 2.2 and supports graphics.

**Vander** Vanderbilt University, H. Denson Burnum, 615-322-2357

**Wash St** Washington State University, Dean Guenther, 509-335-0411, Bitnet: `Guenther@WSUVM1`

**W'mann** Weizmann Institute, Rehovot, Israel, Malka Cymbalista, 08-482443, Bitnet: `Vumalki@Weizmann`

## Screen Previewers

### ▪ Data General MV

**T A&M1** See above for contact name.

### ▪ IBM MVS

**Milan1** See above for contact name.

Drives Tektronix 4014 terminal.

**GMD** See above for contact name.

### ▪ IBM VM/CMS

**HMC** See above for contact name.

DVIview is a previewer written in WEB that can drive VT640-compatible terminals connected to the mainframe by either a 3705 controller or Series-1/7171 protocol converter. It may be obtained by sending $30 (to defray duplication costs), a blank tape, and a return mailer to Don Hosek. The program is still in the developmental stages, and enhancements will be made in the future. The program uses PK files.

**W'mann** See above for contact name.

Previewing on IBM 3279 and 3179-G terminals is provided by DVI82, the Weizmann driver for the Versatec plotter. The program uses PXL files.

### ▪ Siemens BS2000

**GMD** See above for contact name.

### ▪ UNIX

**Adelaide** Programs are on distribution tape.

The DVItoVDU program is capable of driving the following terminals: AED 512; ANSI-compatible; DEC ReGIS; DEC VT100; DEC VT220; Tektronix 4014; and Visual 500, 550.

**Talaris** See above for contact name.

The Talaris driver supports the Talaris 7800 terminal. Tektronix graphics are supported on-screen.

**Utah** See above for contact name.

The Beebe driver family includes a driver for the BBN Bitgraph display.

### ▪ VAX VMS

**Adelaide** Programs are on distribution tape.

The DVItoVDU program is capable of driving the following terminals: AED 512; ANSI-compatible; DEC ReGIS; DEC VT100; DEC VT220; Tektronix 4014; and Visual 500, 550.

**INFN** See above for contact name.

The INFN drivers include support for DEC VT125 and Tektronix 4014 terminals.

**Talaris** See above for contact name.

The Talaris driver supports the Talaris 7800 terminal. Tektronix graphics are supported on-screen.

**Utah** See above for contact name.

The Beebe driver family includes a driver for the BBN Bitgraph display.

**Low-Resolution Printers on Multi-User Systems — Laser Xerographic, Electro-Erosion Printers**

| | Amdahl (MTS) | CDC Cyber | Data General MV | DEC-10 | DEC-20 | HP9000 500 | IBM MVS | IBM VM/CMS | IBM VM/UTS | Prime | Siemens BS2000 | Sym-bolics Lisp | UNIX | VAX VMS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Agfa P400 | | | | | | | | IAM | | | | | | |
| Canon | | | | | Utah | Utah | | | | | | | Canon Utah | Utah |
| DEC LN01 | | | | | | | | | | | | | U Wash1 | NJIT |
| DEC LN03 | | | | | | | | | | | | | | K&S Procyon DEC |
| HP LaserJet Plus | | | | | Utah | Utah | | | | | | | Arbor Utah | Arbor Utah |
| IBM 38xx, 4250, Sherpa | | | | | | | | SLAC Wash St | | | | | | |
| Imagen | Arbor UBC | | T A&M1 | Stanford Vander | Columb. SRI Utah | Utah | Arbor | Arbor SLAC W'mann | | | | MIT | Arbor U Md Utah | Arbor K&S Utah |
| Philips Elpho | | | | | | | | | | | | | | THD |
| PostScript printers | | | | | Utah | Adelaide Arbor Utah | | Arbor | | | | MIT | Arbor Carleton MIT Utah | Utah |
| QMS Lasergrafix | Arbor | U Wash2 | T A&M1 | | | T A&M2 | GMD | Arbor | | T A&M3 | GMD | MIT | Arbor U Wash1 | Arbor GA Tech T A&M3 |
| Symbolics | | | | | U Wash1 | | | | | | | | U Wash1 | U Mass |
| Talaris | | | | | | | Talaris | Wash St | | | | | Talaris | Talaris |
| Xerox Dover | | | | | CMU | | | | | | | | Stanford | |
| Xerox 2700II | | Bochum | | | OSU2 | | | | | | | | OSU2 | |
| Xerox 9700 | Arbor U Mich | | | | | | Arbor T A&M4 | Arbor T A&M4 | T A&M4 | | | | U Del | ACC Arbor T A&M4 |

**Low-Resolution Printers on Multi-User Systems — Impact and Electrostatic Printers**

| | CDC Cyber | Cray | Data General MV | DEC-10 | DEC-20 | HP9000 500 | IBM MVS | IBM VM | Prime | VAX UNIX | VAX VMS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Apple ImageWriter | | | | | Utah | Utah | | | | Utah | LSU Utah |
| DEC LP100 | | | | | OSU2 | | | | | | |
| Facit 4542 | | | | | | | | | | | INFN |
| Florida Data | | | | | MR | | | | | | |
| MPI Sprinter | | | | | Utah | Utah | | | | Utah | Utah |
| Okidata | | | | | Utah | Utah | | | | Utah | Utah |
| Printronix | | | T A&M1 | | Utah | Utah | | | | Utah | Utah |
| Toshiba | | | T A&M1 | | Utah | Utah | | | | Utah | Procyon Utah |
| Varian | | | | | | | | | | | Sci Ap |
| Versatec | U Köln | PPC | T A&M1 | GA Tech Vander | U Wash1 | | GMD U Milan2 | W'mann | LLL | U Wash1 | Caltech2 K&S |

**Low-Resolution Printers on Microcomputers and Workstations — Laser Xerographic, Electro-Erosion Printers**

| | Apollo | Apple Mac-intosh | Atari ST | HP1000 | HP3000 | HP9000 200 | IBM PC | ICL Perq | Integrated Solutions | SUN | Texas Instr PC | VAX-station VMS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Canon | | | | | | | Utah | GMD | Utah | Utah | | |
| Cordata LP300 | | | | | | | Pers | | | | | |
| HP 2680 | | | | JDJW | Pers | | | | | | | |
| HP 2688A | | | | JDJW | | Caltech1 HP | | | | | | |
| HP LaserJet Plus | | | THD Tools | TRC Finl'd | | MPAE | Arbor THD Utah | | Utah | Utah | Arbor | Arbor |
| Imagen | Arbor OCLC | | | | | | A-W Arbor Pers Utah | | Utah | Arbor Sun Utah | | |
| Kyocera | | | THD | | | | THD | | | | | |
| Philips Elpho | | | THD | | | | | | | | | |
| PostScript printers | Arbor | A-W K&S | | | | Arbor | A-W Arbor Pers Utah | | Utah | Arbor MIT Utah | Arbor | Arbor |
| QMS Lasergrafix | Arbor Scan | | | | | | A-W Arbor Pers | | | Arbor MIT U Del | Arbor | |
| Talaris | | | | | | | Talaris | | | Talaris | | |
| Xerox 9700 | COS Scan | | | | | | | | | T A&M4 | | Arbor |

**Low-Resolution Printers on Microcomputers and Workstations — Impact, Electrostatic Printers, and Video Displays**

| | Apollo | Apple Mac-intosh | Atari ST | Cadmus 9200 | HP1000 | HP3000 | IBM PC | Integrated Solutions | SUN | Texas Instr PC | VAX-station VMS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Apple ImageWriter | | A-W<br>K&S | | | | | MR<br>Utah | Utah | Utah | | |
| Diablo | | | | | | Pers | | | | | |
| Epson | | | Tools | | JDJW | U Shef | A-W<br>Milan1<br>Pers<br>U Shef | | | T A&M1 | |
| Fujitsu | | | THD | U Köln | | | | | | | |
| GE 3000 | COS | | | | | | | | | | |
| NEC | | | THD | | | | | | | | |
| MPI Sprinter | | | | | | | Utah | Utah | Utah | | |
| Printronix | | | | | | | Utah | Utah | Utah | T A&M1 | |
| Star | | | THD | | | | | | | | |
| Texas Instruments 855 | | | | | | | | | | T A&M1 | |
| Toshiba | | | | | | | A-W<br>Pers<br>Utah | Utah | Utah | | |
| Video display | Arbor | A-W<br>K&S | THD<br>Tools | U Köln | | | A-W<br>Arbor<br>Pers | UClrv1<br>Utah | Arbor<br>UCB<br>UClrv2 | T A&M1 | Arbor<br>THD |

**Typesetters**

| | Amdahl (MTS) | Apollo | CDC Cyber | DEC-20 | HP3000 | HP9000 200; 500 | IBM MVS | IBM PC | IBM VM | Siemens BS2000 | Sperry 1100 | SUN | UNIX | VAX VMS | VAX-station VMS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Allied Linotype CRTronic | | | | | | | | | | | | | | Procyon | |
| Allied Linotype L100, L300P | Arbor | Arbor | | | | Arbor | | A-W Arbor Pers | | | | Arbor | Arbor | Arbor | |
| Allied Linotype L202 | | | | | | | | Pers | | | | | | Procyon | |
| Alphatype CRS | | | | AMS | | | | | | | | | | | |
| Autologic APS-5, Micro-5 | Arbor | Arbor COS Scan | | Arbor | Arbor | | | Arbor Pers | Arbor | | | Arbor | Arbor | Arbor Interg'ph | Arbor |
| Compugraphic 8400 | | | | | U Shef | | | Pers | | | | | | K&S | |
| Compugraphic 8600 | | | UNI.C | | | | Wash St | Pers | Wash St | | U Wisc | | | K&S | |
| Harris 7500 | | | | | | | | | | | | | SARA | | |
| Hell Digiset | | | | | | | GMD | | | GMD | | | | | |

## DVI Driver Considerations for High-Volume Printing Systems

Thomas J. Reid
Texas A&M University

ABSTRACT: High-speed, high-volume printing systems such as the Xerox 9700/9790 and 8700/8790 Laser Printing Systems present some challenges to DVI driver programs. These challenges stem from the fact that these printers, instead of simply printing master copies to be duplicated elsewhere, can be used to print *all* copies of a document. This article discusses the implications that this has on the design of DVI drivers as well as the services that the drivers should provide.

### The printer as the duplicator

With high-speed, high-volume printing systems, it is possible to use the system to print all copies of a document. This is particularly useful for departments that produce documents that are updated frequently: it becomes economically feasible to print the documents in smaller runs and update them as needed. Thus, "demand printing" is possible. Demand printing means simply printing what you need when you need it.

However, for demand printing to be possible, the output from the printing system must be complete. This means that no manual paste-up can be done and manual page reordering must be kept to a minimum — paste-up and reordering become the responsibilities of TEX and the DVI driver.

Another complication imposed by these printing systems is the ability to print in "duplex" (printing on both sides of the page). With duplexing it is, of course, necessary to print pages on the proper side of the sheet and to pair the correct pages together on the same sheet. Imposition of documents into booklets can be done as can reference card printing. These both require that the driver have special reordering and rotation capabilities.

Finally, the driver should be able to support the paper size/type selection which is offered for the printing system. Selection of sheets from one of a number of trays should be allowed so that covers or divider pages can be printed on a different type or color of paper.

### Paste-up considerations

To support paste-up operations, DVI drivers should be able to handle merging of graphics and mixed orientations on the same page of text. Instructions for doing this should be contained in the user's TEX input file. Since TEX does not have any direct provision for these features, they need to be implemented through a series of TEX macros and `\special` commands.

Details of `\specials` is the topic of another article. However, the capabilities which are important for high-speed printers are:

- *Graphics Merging:* The ability to merge a graphics file into the document and place it at a point on the page specified by the user. The driver should be able to resolve rotations by 90° increments by rotating the graphic itself, or through processing one of four input files for the graphic.
- *Document Rotation:* Since people sometimes want to produce a document entirely in landscape instead of portrait, the driver should be able to rotate the text for an entire document.
- *Rotating a Block of Text:* Rotating a block of text on specific pages involves having mixed orientations on the same page. A good example of this effect is shown in *TUGboat* Vol. 8, No. 2 on pages 166–170; the main text of the page is in landscape while the running title is in portrait.
- *Rotating Words or Characters:* Rotating specific words or characters is useful when using TEX to produce graphs; labels on the y-axis are normally rotated counterclockwise by 90°
- *Simple Graphics:* Simple graphics drawing capability such as diagonal and curved lines should be offered.

### Sorting pages

Although page sorting is not absolutely required, it is desirable for two reasons: it makes the job of page pairing (discussed in the next section) easier, and it eliminates the need for any manual sorting of the final copies of a document.

Page sorting is not as straightforward as it may seem at first. For a typical document, sorting can be done by simply arranging the pages in increasing order by their page number (plain TEX stores the page number in `count[0]` in the DVI file). However, prefatory pages are normally given negative page numbers. These pages need to be sorted in reverse numerical order and placed prior to the main body pages (which have positive page numbers). A question arises on how a page zero should be handled. What should be done to resolve duplicate page numbers?

In a driver that the author is developing, sorting is accomplished by maintaing a "section/subsection" number and a sorting page number.

The sorting page number is simply the absolute value of `count[0]` while the section/subsection number is set to the current section number for positive page numbers or one less than the current section number for negative page numbers. For this purpose, zero is treated as a negative number. If a duplicate page number is found, the current section number is incremented by two to account for the positive and negative subsections in each section. Reordering is then a matter of sorting by the section/subsection number and the sorting page number. While this logic has proven adequate for many applications, there are still some for which it is not appropriate.

*Multi-part page numbers:* Consider a multi-volume document where each volume is composed of parts with multiple chapters per part. An example of this might be documentation for the system services provided by an operating system. The macros used to format the document may place the volume number in `count[0]`, the part number in `count[1]`, the chapter number in `count[2]`, and the page number in `count[3]`. Within each volume, the parts are numbered sequentially from one. Similarly, chapters within parts are numbered from one as are pages within each chapter. Updates are made periodically to the document, but because of its length, major page renumbering is to be avoided. Thus, it is possible to have new pages inserted between existing pages giving page numbers such as 8.1; inserted pages make use of `count[4]`. Further, each volume contains some prefatory pages including a table of contents; the part and chapter numbers are set to zero for prefatory pages and the page number is negative (following the plain TEX standard).

For this application, it is necessary to sort by `count[0]` through `count[4]` with `count[0]` being the major sort key, `count[1]` being next, etc. Since this application uses some special macros for handling page numbers, the following assumptions can be made about the contents of the `count` variables:

- `count[0]`, the volume number, will always be positive.
- `count[1]`, the part number, will be zero or positive. Zero can be treated as a positive number; thus, a section/subsection sort field is not needed.
- `count[2]`, the chapter number, will be positive or zero. As with the part number, zero can be treated as a positive number avoiding the need for a section/subsection field.
- `count[3]`, the page number, will always be positive for non-zero part and chapter numbers and will always be negative otherwise. The value of zero will not occur. Negative numbers are to be sorted by their absolute value. A section/subsection field is not needed since positive and negative page numbers never occur within the same part/chapter.
- `count[4]`, the inserted page number, will be zero or positive.
- Duplicate page numbers (the same values for all five `count` variables) will not occur. Therefore, no provision for handling duplicates is needed.

*Preserve order:* In another application, the TEX input file is created with the pages in the desired sequence. The driver should output the pages in the same order as they exist in the DVI file.

*Generalized sorting:* For a generalized solution, the sorting logic needs to be able to use one or more of the ten `count` values in a user-specified order. Sorting should be permitted by the actual value in the `count` variable or by the absolute value of its contents. For each sort field, an extra section/subsection field may be needed to separate positive and negative numbers or to handle duplicate page numbers. Sorting should also be permitted using a sort field consisting of the sequential number of a page within a DVI file. This would allow the DVI sequence of pages to be preserved.

*Notation for sorting:* Sorting fields can be given for all of the applications presented in this section. The logic implemented in the author's driver corresponds to what might be given by the notation:

```
sort S 0,
```

where "S" means to perform a major sort on a section/subsection number and "0" means to do a secondary sort by the value of `count[0]`. (This notation is not complete since it does not tell how to handle a value of zero; the presence of "S," however, implies that positive and negative numbers should be separated.) Sorting for the multi-volume application might be given as

```
sort 0 1 2 3 4
```

while

```
sort D
```

is used to preserve the page sequence in the DVI file ("D" indicates the field for the sequential page number in the DVI file).

A standard notation is needed by which the user can specify the sort fields. The notation

needs to include provisions for handling positive and negative numbers as well as page zero. Also, for some systems, sorting in descending order may be desirable. The notation should be available both as a `\special` and through a command line option to the driver. Further, guidelines need to be formulated on how the ten `count` fields should be used by applications requiring more than one `count` variable to store a page number.

### Page parity and pairing

Page "pairing" and "parity" are concerns for duplex printers. "Parity" is used here to refer to which side of the sheet a page is to be printed on while "pairing" means which two pages are to be paired together on the same sheet. When using only `count[0]` for the page number, the rules are easy: odd pages on the front with pages $2n-1$ and $2n$ (where $n$ is an integer) placed on the same sheet. Page number zero again causes a slight problem: since it is typically used for a title page, it should normally be considered odd and placed on the front of a sheet.

**Page parity.** In the multi-volume application of the previous section, `count[0]`, `count[1]` and `count[2]` (the volume, part and chapter numbers) should be disregarded for page parity consideration; only the page number within each chapter is to be used. However, the need for inserted pages requires that both `count[3]` and `count[4]` be treated as the page number. Parity can be determined by adding the two values together: page 8 $(8+0=8)$ is even and should be placed on the back of a sheet; page 8.1 $(8+1=9)$ is odd and should be placed on the front; page 8.2 $(8+2=10)$ is even, etc.

In the application where the DVI page order is to be preserved, it is assumed that the user has made provisions for page parity — the sequential page counter field "D" can be used to determine the page parity. The driver would assign values to the "D" variable beginning with one and incrementing by one. Hence, the values will be alternating odd and even numbers corresponding to alternating fronts and backs.

*Notation for page parity:* Using a notation similar to that used for sorting,

```
parity 0
```

becomes the default rule while

```
parity 3 4
```

is used in the multi-volume application and

```
parity D
```

is used when the user has made provisions in his or her TEX source file. A standard notation should be developed for page parity along with the sorting notation. This standard should have a provision for page zero: is it even or odd?

**Page pairing.** In the driver which the author has implemented, page pairing is done by selecting one or two pages from a sorted page list. The following rules are used to select the page or pages:

1. Select the next page in the list.
2. If the number is even:
   a. Put a blank page on the front of the sheet,
   b. Put the selected page on the back of the sheet; and
   c. Return to step 1 for the next sheet.
3. If the number is odd:
   a. Put the selected page on the front of the sheet;
   b. Examine (but do not select) the next page in the list.
   c. If it is even and has a page number one greater than that of the page on the front, select the page and put it on the back of the sheet then return to step 1 for the next sheet.
   d. If it is even and does not belong to the page on the front, or if it is odd, put a blank page on the back of the current sheet and return to step 1 for the next sheet.

This algorithm works fine if only one `count` variable is used for the page number. In the case of the multi-volume document, both `count[3]` and `count[4]` need to be considered for page pairing. For example, page 7.1 could be placed on the same sheet with page 7 while page 7.2 and 8 share the next sheet. Thus, step 3c needs to be generalized for multiple page-parity variables.

### Imposition

Imposition can take on many different forms. In printing books, 32 consecutive pages are combined on a sheet of paper which is then folded and cut. The resulting group of pages is called a signature. Other forms are common which place 16 pages per signature. Since high-speed printing systems typically use $8\frac{1}{2}'' \times 11''$ paper, 32 or 16 page signatures are normally not used. Instead, imposition is used for booklets that are folded once to $5\frac{1}{2}'' \times 8\frac{1}{2}''$ and reference cards that are folded two or three times. Both require additional page ordering capabilities as well as the ability to rotate text.

These two forms of imposition typically result in final pages which are smaller than the normal size that TEX produces. It is the responsibility of the user to set the proper values for `\hsize`, `\vsize`, `\hoffset` and `\voffset` and to select fonts that are of an appropriate size for the smaller page. The driver, however, should handle the rotation of a page from portrait to landscape or inverse landscape since it is difficult for the user to know in advance which side of the sheet a page will print on.

**Booklets.** To impose a document into a booklet which is folded, the driver needs to place two pages side-by-side on the front of a sheet in landscape (rotated by 90°) and two pages side-by-side on the back in inverse landscape (rotated by 270°). When the driver performs this rotation, it needs to compound any rotation specifically requested by the user. For example, if the user asks for a table to be rotated by 90° (i.e., landscape on a portrait page), the driver should print the table in inverse portrait ($90° + 90° = 180°$) on the front of a sheet or portrait ($90° + 270° = 360° = 0°$) on the back.

It is also necessary to reorder the pages so that after the set of sheets comprising the document (that is, a "signature") is folded, the pages will be in the proper sequence.

The author has implemented this by building a list of the pages to appear in a signature. The list entries are set a pair at a time using the normal duplex page pairing logic. After the list is built and padded with blank pages to contain a multiple of four pages, entries are selected four at a time beginning from the center and working toward the two ends. The four selected page entries are printed on the front and back of a sheet.

**Reference cards.** Reference cards normally have three panels on each side although they can have more if larger card stock is used. As with imposition, it is necessary to print the front side in landscape and the back side in inverse landscape. Special reordering is also necessary, but it is simpler than that for booklets. The author proposes two such ordering arrangements. Both involve selecting six pages (for a three panel card or eight pages for a four panel one) from the sorted page list. The set can then be printed with pages 1, 2 and 3 on the front and 4, 5 and 6 on the back, or it can be printed with 2, 3 and 4 on the front and 5, 6 and 1 on the back.

### Page selection / document updates

On proof-mode printers, page selection is useful because it allows an author to print a subset of an entire document. In a demand printing environment, page selection is also important because it may be desirable to print many copies of one section of a longer document. A special case of page selection should also be available which allows only updated pages to be printed. Updated pages may be indicated by a non-zero value or a date in one or more of the `count` variables. In the multi-volume application, an update package consisting of only revised pages would be much preferable to reprinting an entire chapter, part, or volume.

However, when producing duplex print, it is generally necessary to print both pages which share a sheet even though only one of them may have been requested or updated. This is easily accomplished by producing a sorted page list containing all of the pages. Each page list entry will have a flag indicating whether or not it has been selected or updated. The rules for page pairing shown earlier can then be modified by adding a fourth step which tests the flag(s) of the selected page(s) to see if the sheet is to be printed. If either or both flags are set, the sheet is processed in the normal manner. Otherwise, the selected page or pages are discarded.

### Media

The Xerox 9700/9790 and 8700/8790 provide two input trays and allow the user to select paper from one tray or the other. Thus, it is possible to select main body pages from the "main" tray while pulling covers or dividers from the "aux" tray.

Many centers offer a variety of types and weights of paper for their printing systems. At the author's site, a selection of pre-drilled and non-drilled white paper, plain paper and cover stock in several colors, bond paper, and label stock are offered. It is possible for the user to request that two different types of paper be placed in the two feed trays.

It would be advantageous for paper type codes to be placed in a TEX file (in the form of `\specials` and command line arguments). The ability to select from one of a number of trays is also desirable. The DVI driver should then translate these type codes into a form for the printer according to the operational policies of the center. This would, of course, be site dependent since operational policies are non-standard.

Another consideration for media is the ability to print paper sizes other than $8\frac{1}{2}'' \times 11''$. This

requires some changes in origin placement but is generally fairly simple.

### Future printing systems

Although high-speed, high-volume printing systems do not yet offer color printing capabilities, future systems likely will. Through TEX macros and `\special` commands, the user should be able to control the color of the text and/or graphics on a page.

Binding is another feature which may be offered by newer printing systems. If a printer has options for user-controlled binding, the DVI driver should be able to support them.

Any standards or guidelines which are developed concerning `\specials` and DVI drivers need to be open-ended so that capabilities of future printing systems can be incorporated into them.

### Conclusion

Demand printing applications made possible by high-speed, high-volume printing systems make additional demands of the document preparation software. TEX represents a sound system for handling the typesetting aspect of the document preparation. However, DVI drivers need to take on a more prominent role in the document preparation cycle by providing the non-typesetting capabilities needed for support of these printing systems.

Since the significance of the responsibilities of the DVI driver are raised to the level of those of TEX, more attention to the DVI driver is needed. While TEX represents a standard for typesetting, no such standard currently exists for DVI drivers — each driver author is left to his or her own imagination. Efforts are presently underway to propose a standard for `\specials`; similar efforts are needed to formulate standards and/or guidelines for DVI drivers.

## `\special` issues

Glenn L. Vanderburg and Thomas J. Reid
Texas A&M University

ABSTRACT: As more and more DVI translation programs appear, it is important to have a standard set of `\special` commands, and a standard format for those commands, so that attention can be focused on the documents being prepared rather than on the printer being used. This article contains a discussion of the various problems associated with specials and describes a format and set of specials which could serve as the foundation for a standard.

One of the most farsighted and useful features of the TEX language is the `\special` command. In acknowledging the incompleteness of TEX and of the DVI format, Professor Knuth gave us the mechanism by which we could do much to extend the language and adapt it to changing needs and capabilities.

In *The TEXbook* ([16] page 229), Knuth has this to say:

> Whenever you use `\special`, you are taking a chance that your output file will not be printable on all output devices, because all `\special` functions are extensions to TEX. However, the author anticipates that certain standards for common graphic operations will emerge in the TEX user community, after careful experiments have been made by different groups of people; then there will be a chance for some uniformity in the use of `\special` extensions.

The time has come to define the standards which Knuth proposed. The TEX community is maturing rapidly. DVI drivers exist for many printers, but most such programs are still being developed. Powerful page-description languages now exist, and TEX users are eager to exploit their capabilities. Many drivers have implemented some specials, and some lessons have been learned. Questions and articles in *TUGboat* and *TEXhax* have demonstrated the kind of functionality which is required. A standard for the format (and function) of special commands is important, and it will be most useful now, while so many drivers are being developed. The authors have been considering this problem for more than a year, in connection with the development of a DVI driver for the Xerox 9700 family of printers, and would like to present their ideas and suggestions. Many of the ideas in this article are due (at least in part) to Dr. Bart Childs, who has spent a great deal of time considering these issues and sharing his ideas. This is not meant to imply that he endorses the recommendations

here, as his original ideas have been modified considerably. His input was important, however.

We also were influenced by Robert McGaffey [18], Don Hosek [12], and others too numerous to name; we've seen many drivers and talked to many users about their needs. *TUGboat* and *TEXhax* have been invaluable aids, with articles and discussion about specials and the problems that TEX users face.

We hope that the TEX community, and specifically the TEX Users Group (TUG), can adopt a standard which will detail the recommended behavior for all DVI translation programs. It will not be necessary for every driver to implement each special command in the standard, but each driver should support a minimal subset of the standard specials, so that users will know what to expect with regard to certain very basic operations.

The remainder of this article considers problems of function, form, and use of specials, along with the problem of device-dependence.

## Guidelines and Goals

There are certain guidelines (or design principles) which should govern the standard, and certain goals to drive it.

- Any DVI file should work with any driver as well as possible. Unrecognized specials should not cause problems, and a given special should produce equivalent effects (where possible) on all printers. In particular, this means that specials should not be tagged with driver or printer names.
- No device-specific information should be included in special commands, except where absolutely necessary.
- Keep specials general. When there are frequently used special cases, implement them as aliases for a more general special (although ideally those special cases should be accessed by a TEX macro to generate the special).
- Text should be handled by the same specials that are used for other graphical objects. This will make the generation of special effects with specials much easier. PostScript is a good example of how successful this approach can be [4].
- The DVI philosophy of compactness should be followed ([15], section 584).
- Keep the format of specials simple and easy to parse. Don't make the driver do extra work that could be done more easily by TEX. This is not to say that "if it can be done in TEX, you don't need a special for it." However, the work should be done where it's easiest.
- It is desirable for TEX users to be able to produce output in its final form (camera-ready, for instance) directly from the DVI driver. Where this is not possible, it should be because of inherent printer limitations or because the standard has not been fully implemented, rather than because of limitations in the standard itself.
- Design for a very capable printer and driver. This will encourage the development of such printers and drivers, by accenting the deficiency of those which cannot (or do not) support the whole standard.
- All applicable command-line options *must* be accessible via specials. Frequently, TEX users must print new copies of a document which was written years ago. If that document required special command-line options to print correctly (`landscape`, for instance), it might take some experimentation to rediscover those options. It would be extremely convenient to have those options encoded in the DVI file. Options to specify printing of multiple copies or to select pages from the document would not be good candidates for such specials, however.
- There should be a convention for global specials.
- A conforming driver should not be tied to any one format (or set of formats) for graphics files.
- The DVI driver should be a good DVI driver and little else; it should not attempt to be a graphics interpreter or translator.
- Don't underestimate the number, variety, or complexity of users' requirements. Make provisions for extension of the standard.

It may seem that some of these goals are contradictory. It is likely that in some cases it will prove impossible to satisfy all of them completely. They are all desirable goals, however, and it should be possible to meet them with a little work, and the standard will be a better one for the effort.

## Types of specials and their functions

It seems appropriate to begin by discussing the functions that will be required, since that will have a significant effect on other aspects of the standard.

**Global specials.** Much has been said and written about the implicit restriction on specials affecting pages other than the one on which they occur. While the random ordering of pages in the DVI file does imply that this cannot occur, the authors see

no need for it to be a hard and fast rule. Some convention for global specials should be developed, and then they should be used with care. The DVI format permits specials to occur before the beginning of the first page and between pages ([15] section 588). Unfortunately, TEX82 does not put any specials there. There are some reasonable possibilities, however. Specials which occur at the beginning of the first page in the DVI file and which have a `global` prefix could have global effects. Certain other specials could have effects on pages which succeed them in the final sort order; they would merely have to be used with care. A policy such as that adopted by Adobe Systems, Inc., toward PostScript is advisable [3]: they advocate keeping pages self-contained, but they do not enforce this. The ability to make something on one page affect other pages is there, but Adobe advises caution when using it. If the sort order is changed drastically or if a single page is extracted from the file, chaos will ensue. These are unusual cases, however, and the usefulness of global specials makes the danger worthwhile, in the opinion of the authors. An alternative approach is to have an optional companion file for the DVI file which would contain specials which could have a global effect.

**Text and graphics manipulation.** These specials will affect a region of text or graphic objects, some of which may be produced by other specials. Note that "region" refers to a region of the DVI file, rather than any particular area of the physical page. We will use "area" to specify a portion of the physical page. We should also take this opportunity to point out two other important points: the order in which transformations are applied is important, and some decision needs to be made on whether delimited specials should be allowed to nest. To permit special effects, the standard should provide for all of the common two-dimensional transformations, especially since they are easily accessible in modern page-description languages.

*Rotation:* A special for rotation should be capable of rotating graphics or text, up to a whole page. Of course, when graphics or text are stored in bitmap form, it is only feasible to rotate in 90° increments (and sometimes that isn't possible either, depending on the output device). But where it is possible to rotate, the portion of the page to be rotated could range from a single character to the entire page. As an example of how complex the design of a special can be, we will attempt to specify a general rotation special in its entirety.

First, the driver needs to know what to rotate. We could limit the special to rotating a single box (an hbox or a vbox), and use a single special which would rotate the next box in the DVI file (box structure can be inferred from the levels of the *push* and *pop* commands in the file [15][17]). This has two disadvantages, however. It makes extra work for the driver, keeping track of pushes and pops, and it also limits the scope of the command. A rotation special defined in this way would not be able to reliably rotate an entire page. In Plain TEX, a page is placed in a box by the output routine, and then shipped out. The special would go inside that box, and could not be placed outside of it. Of course, we could define the special as rotating the immediately enclosing box, but this is hard to control and it runs contrary to a user's intuitive understanding of the special. So we choose to have a delimited special, begun by, say, `rotate` and delimited by, say, `etator`. In this way, we can specify an arbitrary region to be rotated in a very general way. Whole pages can be rotated easily, and we can handle this in a global fashion by modifying the output routine to use `\everybop` and `\everyeop` token-list registers, containing commands which would be placed at the beginning and ending of every page. In such a case, the `etator` would not be strictly necessary; the region of a delimited special would be bounded by the end of a page (except possibly if it had been declared `global` — see *Global specials*, above).

Next, the driver needs to know the pivot point (the point about which to rotate). It is easy to just have all text rotated around the current coördinates, but that is needlessly restrictive. Such a restriction would result in a lot of clumsy TEX hackery to achieve other effects. Therefore a user should be able to specify an offset, using any of the standard TEX dimensions (note that "true" dimensions must be supported also). Macros can be constructed to expand to the offset of the upper-right-hand corner of a box, or the center, etc., so that this scheme would not be too difficult to use. One should also be allowed to specify exact coördinates, or to use a predefined point (see *Plotting*, below).

Finally, the driver needs to know the angle to rotate through. In the simplest case, it could be a number which would represent degrees. Ordinarily, it is not important to know from what direction an angle is measured; the default case is 0°, and angles are measured counterclockwise from that. It would help, however, to settle on a "text direction" concept to simplify things for users and for the authors of drivers. Angles should be measured

counterclockwise from a horizontal line. This is intuitive, and it also makes the transformation easier for the driver, because it simplifies the mathematics; the transformation matrix to be used becomes

$$\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

for the angle $\theta$. If the angle is in units of degrees, it may be necessary for the driver to convert it to radians before computing the sine and cosine.

When the angle is given in its component parts (i.e., two numbers), the two numbers should be assumed to be related to the sine and cosine of the angle. Thus, the angle $\theta$ can be determined from the components $x$ and $y$ by one of the following relations:

$\theta \leftarrow \arctan y/x$; for $x > 0$

$\theta \leftarrow \arctan y/x + 180$; for $x < 0$

$\theta \leftarrow 90$; for $x = 0$ and $y > 0$

$\theta \leftarrow -90$; for $x = 0$ and $y < 0$

error; for $x = 0$ and $y = 0$.

This gives the angle $\theta$ in the range $[-90, 270)$. If it is desired to have the angles in the range $[0, 360)$, one can say:

$\theta \leftarrow \theta$; for $\theta \geq 0$

$\theta \leftarrow \theta + 360$; for $\theta < 0$.

In working with rotations, the angle itself is not generally useful; the sine and cosine are the important values, as shown in the transformation matrix above. If an angle is given in component form, it is possible to calculate the sine and cosine more efficiently using the following relations:

$d \leftarrow \sqrt{x^2 + y^2}$

**if** $d = 0$ **then** error

$\cos\theta \leftarrow x/d$

$\sin\theta \leftarrow y/d$.

So in order to include all of the necessary information, the `rotate` special must have the following form (or something similar):

`rotate` *hoffset voffset angle* ⟨*stuff*⟩ `etator`.

It should be obvious that a well-designed special is the result of a lot of work. Many factors need to be considered: desired capabilities, the user, TEX, the DVI format, implementation details, and printer capabilities.

There is one further note about the direction of text. Here, we have decided that angles should be measured counterclockwise from the positive $h$ axis. Readers of right-to-left languages [17] might not choose that, however; they might choose clockwise from the negative $h$ axis. At least some consideration should be given to making the "direction of text" equal to the direction in which the current point is displaced when printing it. This would complicate the implementation, but it might make the mechanism more versatile.

*Scaling:* If the printer supports it, there should be a special for scaling of text and other graphical objects (whenever they are not stored as bitmaps, of course). Possibly an offset would be needed here, also; do we want all of the expansion to be away from the current reference point? The scaling special should also be delimited, and it should support scaling by factors and also to a certain explicit size. Independent horizontal and vertical scaling should be possible. This would distort pictures in an undesirable fashion, but it would leave the data content of many graphs intact, while fitting them into a space of arbitrary size.

*Translation:* Translation would be helpful in placing graphic objects on the page. For example, suppose one wanted to use graphics inclusion to print a form which would overlay the entire page. Horizontal and vertical offsets should be the arguments here. Whether or not it should be delimited is a matter for debate; it can be strongly argued that any objects actually created by TEX (i.e., not by the driver itself) should be positioned on the page using TEX.

*Reflection:* Reflection requires knowledge of a region and an axis. The axis could be specified by two points or by one point and an angle. Note that this special is not strictly necessary, as reflection can be accomplished by a combination of other specials.

*Clipping:* The driver should be able to do simple clipping. This would be very useful when including graphics such as the METAFONT proofs produced by `GFtoDVI`. That program puts a title on the top of the page, and there is no option to suppress the title. Therefore it would be nice to be able to tell the driver to print a certain area of the page, and print it here. The standard could only permit rectangular clipping paths, although a more general mechanism of path specification might be necessary for a `fill` special (see *Plotting*, below). Note that the user should never assume that characters will be partially clipped, even though page-description languages such as PostScript permit it. The standard should specify the handling of characters which fall partially within the clipping area, and it should also permit specification of whether to clip inside

or outside of the clipping area. A generalization of an "off-the-page" check would make this relatively easy to implement.

*Color:* Printers are beginning to incorporate color. The PostScript and Interpress [21] page-description languages have facilities for controlling color, and Adobe's Illustrator program [2] (which generates PostScript) can generate color documents automatically. The standard for specials needs to provide a way to access color facilities.

If color is to be accessed via a special, a complete range of colors must be supported. Gray scales should be available. Most importantly, a common color model should be used. Color models are difficult to understand; the user shouldn't have to learn a new one. The common color models are described in [9] on pages 611–623. From page 611:

> Three hardware-oriented models are the RGB (red, green, blue), used with color TV monitors; YIQ, which is the broadcast TV color system; and CMY (cyan, magenta, yellow) for color printing devices. Unfortunately, none of these models are particularly easy for a programmer or application user to control, because they do not directly relate to our intuitive color notions of hue, saturation, and brightness. Therefore, another class of models has been developed with ease of use as a goal. Several such models exist; we shall discuss only two: the HSV (hue, saturation, value) and HLS (hue, lightness, saturation) models.

RGB and YIQ were designed for luminous devices, CMY for hardcopy devices, and HSV and HLS for users. One of these models will almost certainly be understood (in some form) by the output device, and the driver will have to map whatever is specified in the special to the printer's model. Fortunately, there are standard mappings between all of these models; in fact, Foley and Van Dam give the code in their book. None of the algorithms are particularly complex. It seems reasonable, therefore, for the standard to support all five models. Individual drivers may not support all of the models, but they should at least support the printer's native model, and it would not be difficult to support all of them.

Drivers for black-and-white printers could render color in grayscales by mapping to the YIQ model. YIQ was designed for downward compatibility with black-and-white TV; The Y component can be used alone to represent grayscale.

It is important to also permit specification of background color. This would permit setting white text on a black background. This special should specify an area in the same manner as the `clip` and `fill` specials.

**Graphics inclusion.** This is really one of the stickiest areas of a special standard, because it contains most of the device-dependent details. We should resist the temptation to incorporate scaling, translation, and clipping parameters into these specials; it would be better to use the general specials described above for such things.

*DVI inclusion:* This one keeps free of device dependency, but it's certainly not easy to implement. Fonts in a DVI file are numbered [15], so font numbers and coördinates need to be kept separately for different DVI files [20]. It would be really nice, however, to be able to include pages from other DVI files into a document, or even parts of pages. It also gives access to the magnification feature for scaling a page to be included. This could be very useful: *The METAFONTbook* could have been produced with no manual pasteup of any kind. The proofs from METAFONT could have been included in this way. In our opinion, this is the only kind of file inclusion for which the DVI driver should act as an interpreter in any way.

*Standard graphic formats:* Other programs should be used to translate the files to printer-specific format. This keeps the driver from being too complicated, and it allows easy extension to other formats. In addition, the other programs could create a companion TEX file for the graphic with TEX commands detailing the size of the graphic, etc. It could even place those commands at the beginning of the graphic file itself; an `\endinput` command would cause TEX to exit from the file before it hit the graphic information, and the driver could strip those commands out when it read the file. The preferred method, in the opinion of the authors, is to have a companion file.

*Printer-specific formats:* This would seem to be the easiest task. We should just include the file and go, trusting that the file itself is error free. If the file is properly encoded, this should work on almost all printers. A special conversion program to properly "encapsulate" the file would be a fairly simple task. There is, however, at least one complication, which is the subject of the next section.

*Orientation-specific files:* On some printers (e.g., the Xerox 9700 family), graphics files are orientation-dependent. A graphic created to be printed in landscape will not work correctly in inverse landscape. This can become a problem,

especially when imposing pages. Suppose we are imposing pages to produce a booklet which can be folded and stapled straight from the printer. Pages on the front of a sheet are in landscape, and pages on the back of a sheet are in inverse landscape. When the document is being formatted we do not know which side of the sheet the graphic will finally fall on; this may even be dynamic, and we may print the document in simplex for proofing to speed turnaround. Admittedly, this is a printer limitation, but we should provide for it. It would probably be wise to provide an `if else` special. If so, it should be generalized to test not only orientation, but other conditions and user-defined values, as well.

**Direct printing features.** Some of these specials will also be device-dependent, and site-dependent as well. Defining such site-dependencies should be done in a configuration file, separate from the other source files if possible.

*Paper types:* Many computer centers offer multiple paper types. Usually they are referenced by a special code. This special should take multiple arguments, and the driver should map to a paper type code for the particular site. At a center where the users are familiar with the codes, the argument would be the code itself. At other places, users might specify `green cardstock`, and the driver would substitute the proper code (if, that is, the driver is told to go ahead and print the file). Note that there should be a way to specify which type of paper should be put into which input tray, for printers with multiple trays.

*Tray selection:* Many printers now have multiple paper input trays. The Xerox 9700 has two, the DEC PrintServer 40 has three, and even the Apple LaserWriter has two (if you count the manual feed). It is extremely convenient to be able to specify that, for instance, the cover and chapter dividers should be pulled from tray 2, so that no hand collating is necessary. This is obviously printer-dependent.

*Other options:* The direct printing features demonstrate the need for extensibility. Some printing centers may offer binding. Should there be a special to control binding? Xerox' Interpress page-description language [21] has facilities to specify binding, cutting, etc. The *Adobe Systems Document Structuring Conventions* [1] also define notations which control such things. It should be clear that at this time it is extremely unlikely that all possibilities can be anticipated.

**Page selection and ordering.** This will be a very difficult set of specials to design. Even simple sorting is a sticky problem. Is page zero odd or even? Positive or negative? Should there be a special to control that? When more sophisticated capabilities are desired, there are even more issues to consider. These options would be most useful on the command line, but they would also be useful as specials.

*Sort keys:* At the beginning of every page, the values of `\count0` through `\count9` are included in the DVI file. It would be very convenient to be able to specify which of these values would be used as the primary sort key, which would be the secondary sort key, etc. In this way, page order could be controlled explicitly.

Sorting of pages should be allowed using all ten of the `\count` registers as well as at least one special register. One special register is defined as the pages are being scanned in the DVI file. Its value is the sequential number of the page in the DVI file, the first being 1, the second being 2, etc. The backpointers in the DVI file could still be used to skip around in the file if this register was not going to be used for sorting. Some mechanism for handling duplicate page numbers would be necessary; ideally a document would not have any duplicates.

As an example of how different sort fields might be used, consider a document with the page number made up as follows:

`\count0` is the current chapter number;
`\count1` is the current section number within the chapter;
`\count2` is the page number within the current section and chapter;
`\count3` is used for any inserted pages.

The pages for this document need to be sorted by all four of those registers, with `\count0` as the most significant sort key. It is important to note, however, that the significance of the registers should be variable. Also, we might not always want to sort groups of pages with negative sort keys by the absolute value of the key. It might even be desirable to sort in descending order, especially for printers that stack their output face-up.

*Page parity:* Page parity controls whether or not a page should be placed on the front or back of a sheet when printing on both sides of a page. (We will refer to such printing as "duplex" printing, and to printing on one side of the page as "simplex" printing.) In the simplest case, odd page numbers will go on the front, and even page numbers will go

on the back. Readers of Japanese or Hebrew may feel differently, however.

Determining the "parity" of a page is fairly easy if only one `\count` register is used to give the page number: the parity is simply the parity of the `\count` register (with provisions for the value zero and right-to-left languages). However, when more than one `\count` register is used, the problem is more complex.

Again, consider the sample document in the previous section. Four `\count` registers are used to contain the page number. Chapter and section numbers have little to do with which side of the sheet a page should print on. Therefore, `\count0` and `\count1` should be ignored in determining page parity. But this still leaves `\count2` and `\count3`. The actual parity of the page should be based on the sum of these two registers. This is because page 1.2.2 should fall on the back of a sheet (with page 1.2.1 on the front) while page 1.2.2.1 falls on the front of the next sheet. Similarly, page 1.2.3 goes on the front and page 1.2.3.1 should be printed on the back of the same sheet. A special to control parity should be able to make use of the same registers used for sorting.

*Page pairing:* Another aspect of page parity is choosing which two pages should be placed on opposite sides of the same sheet. This determination is made by selecting pages from the sorted page list. The algorithm takes one or two pages from the page list and defines the page or pages to be placed on a single sheet. As long as the page numbers remain consecutive, this is not a big problem, but it can get sticky. Should page six fall on the back of page one if it occurs next in the final sort order?

*Version numbers:* Imagine a user documentation division of a computer corporation. They might have files called `usermanualV1.tex` and `usermanualV1--1.tex`. They contain version 1 and version 1–1 of the usermanual, respectively. The first should take care of itself, but the second might pose some problems. The group uses macros to mark changes that are inserted. Those macros use some plotting specials to insert changebars into the text, but that is not our concern right now. Two different kinds of behavior will be desired from that file. The entire version 1–1 manual will need to be printed for distribution to new customers, but only the changed pages will be needed for distribution to old customers as updates. Those pages should be marked somehow by a special. One possibility is to allocate a count register (say, `\count9`) as a change flag, and have a global special and command-line option which instructs the driver to print the page only if that register has a certain value. The register to use should not be hard-coded into the special. Perhaps a `dontprint` special could be used in conjunction with the aforementioned `if else` pair. Note that for duplex printing the driver will have to also print the page on the other side of the sheet, even if that page was not changed. The nature of this special makes it easier if it occurs immediately after the beginning of the document.

**Page manipulation.** These specials are for describing the page to the driver, moving it around, etc. These are good candidates for `global` treatment.

*Page size:* This special would be used to define the size of the sheet of paper to the driver. Many printers are equipped to handle multiple sizes of paper, so that should be settable via a special. A default setting should be defined in a configuration file, so that users in Europe can use A4 settings, for example.

*Imposition options:* This is used to place multiple pages on a single sheet which can later be folded or cut to make a booklet. This special needs to be very general; for example, in magazine production, eight pages are placed on each plate, so that each sheet coming off the press contains sixteen pages (eight on each side). This large sheet is then folded in a weird way, stapled with zero or more similar assemblies, and trimmed to make the final magazine [13]. The imposition special should allow specification of the number of pages per sheet, their placements on the sheet, and their orientations. Note that scaling should not be the responsibility of the driver; it is the user's responsibility to make the pages the correct size and to set the margins. Note also that the sort order might be different depending on whether the sheets are to be folded together or cut, punched, and put into a binder.

*Overprinting:* It might be very useful to provide for superimposing two pages, one on top of the other. This would be an easy way to produce a portrait page (with portrait headline and footline) containing a landscape table. It should probably not be a separate special, though. If a special to include a page from another DVI file is implemented, it should be general enough to permit including another page from the same DVI file onto the current page. The translation special could be used for positioning.

*Full-page orientation:* Although the rotation special we mentioned above is general enough to

rotate an entire page, a special which is devoted to rotating the entire page might be worthwhile. It would be much easier to use as a `global` special. The same applies for a full-page translation special, for margin adjustment.

**Plotting.** TEX users really need some sort of simple plotting interface. Using a separate graphics package to create simple pictures or graphs and then including them in the TEX document is often too difficult. A simple plotting interface would ease this difficulty, and allow users to produce figures like those produced so easily with the *pic*, *grap*, and *chem* preprocessors for *troff* [6][19]. Plotting functions would also make TEX more suitable for special applications such as music typesetting [10][11]. (Incidentally, anyone not familiar with *grap* [7][8] should take a little time to study it; it is a fantastic example of a well-designed little package. It is elegant, very easy to use, has reasonable default behavior, and can be persuaded to do some fantastic things.)

*Point definition:* This idea came from drivers produced by ArborText. We saw it in a driver for the DEC LN03, written by Flavio Rose. The driver maintains an array of point coördinates (initially all zero). A special is used to define one of those points at the current location. That point can later be used as the location of a dot or the end of a line. The great thing about this is that it allows the position of a point on a page to be located by TEX's formatting (which is very difficult to do without some sort of special), and that location can then be used on other pages. Admittedly, no great application for this is immediately obvious, but it does seem useful, and it is so difficult to do with just TEX and the other specials that this special seems worthwhile. It also violates the implicit restriction on specials affecting other pages, but it is a simple fact that sometimes we will need just that (see *Global specials*, above).

*Splines:* This special would be used to specify an arbitrary curve between two points (which can be explicit coördinates, offsets, or points defined by the `pointdef` special). One should be able to specify a curve from the current point to another, and also between two arbitrary points. The shape of the curve should be determined by two control points. Specials to define line thickness and cap shape should also be available.

*Geometric shapes:* For ease of implementation, it might be desirable to supply straight lines, circles, and other common shapes just as special cases of the spline-drawing facility, but there are some good reasons not to. First, computation of the shapes will be faster if there is a special facility devoted to them. Second, there are special techniques for digitizing straight lines which do not work as well for curved lines. Offering a separate special for straight lines will permit special handling.

*Filling:* This should be restricted to filling with a solid color. Any filling which requires a complex pattern should be done as a separate graphic which would be included. Some way of defining a closed path will be necessary, as will a convention for handling what METAFONT calls "strange paths" [14].

**Miscellaneous specials.** As one might expect, there are some specials which do not fit easily into categories.

*Communication with the user:* Just as TEX has the `\message` command, it might be desirable for the DVI driver to have a `message` special. In addition, drivers should optionally warn users of unrecognized specials instead of ignoring them. There should be a special to invoke that option. It would be reasonable to make that special the only one which *must* be supported by a conforming driver.

*Job logging:* Several drivers keep logs in much the same fashion as TEX does. It might be advisable to offer specials to control this. Logging could be enabled or disabled via a special. In addition, one could specify whether the driver would keep a separate log or append to the TEX log file.

*Literal commands (`\special{special}`):* For many printers, passing literal commands straight through makes no sense; the effect would be entirely unpredictable. For some printers, it would be possible, though. The wisdom of providing such a special is debatable (especially if the other specials offer a great deal of functionality), but it should be considered. Note that this is the most device-dependent of all specials.

*DVI format:* There is already at least one extension to TEX which makes use of some of the unused commands in the DVI format: the `TeX-XeT` processor described in [17]. Perhaps such extensions should use a different DVI identification byte, but it is also conceivable that those variations could be identified by a special. Either way, a single driver could support multiple extensions to TEX. Incidentally, the authors prefer using the DVI id byte for this.

### Accessing specials from TEX

Specials should be generated by macros whenever possible. Users should not have to learn to use another user interface; they should just have a few new macros. `\landscape` could be a macro meaning that the current page should be rotated about the origin and then translated by *pageheight* − 2 *inches* in the $-h$ direction. This would simplify the user interface greatly.

Some extra macros would also simplify the `global` problem slightly. `\everybop` and `\everyeop` token-list registers could be defined which would be placed by the output routine at the beginning and ending of every page. Also, a `\globalspecial` register could be used. If this register was defined before the first `\eject`, the output routine would ensure that the contents occurred first on the very first page, before any *set* or *put* commands. This would require only trivial modification to the output routines, and the benefits would be enormous.

### Handling device-dependence

As much as we try to avoid it, some specials are going to be device-dependent. This is a hazard of using specials. The trauma could be reduced, however. METAFONT has a nice little feature in its ability to handle "modes" [14]; different modes for different printers. Such a facility could be implemented in TEX with macros. It would rarely be used, but it would help with certain device-dependent specials. Typing `\mode{X9700}` would define certain special-generating macros to fit the device. Modes would be defined in a `local.tex` file, just as METAFONT's modes are usually defined in a `local.mf` file. This would greatly simplify the process of producing proof copy on one device and then moving to another device for the final output. Note that it may be advisable to name the modes after the specific driver rather than the printer.

### Miscellaneous recommendations for driver behavior

Robert McGaffey's article, "The Ideal TEX Driver" [18], offers some excellent recommendations for the behavior of DVI drivers. Some of his suggestions (those regarding the handling of missing fonts) are also good candidates for handling with specials. We would like to offer some additional recommendations.

- The driver should be able to read a configuration file on startup, so that the user can include certain default options there. McGaffey mentions this, but the idea is important enough to merit discussion here as well. Possibly a system-wide configuration file should be read first, and then the user's. This might be one solution to the problem of global specials.
- The driver should also be able to look at certain environment variables defined by the user. Where the authors work, default paper type and destination are set in this way.
- Where there is complex logic (such as sorting logic), it is inevitable that a situation will surface where that logic is inappropriate. The driver should provide an option to turn off that complex logic and go back to the brute force method.
- In the configuration file or an environment variable, the user should be able to specify a "path" for the location of TFM and bitmap files, so that personal METAFONT-generated fonts can be used with ease. This mechanism should be able to deal with the various complicated and contradictory conventions for naming and storing such files. The authors are working on a mechanism for a driver driver which apparently will permit such specification in a very general way.
- There is naturally some confusion about how to handle the PostScript fonts and their ability to be scaled. TFM files for those fonts should probably be generated assuming a default size of 10 points, to be consistent with standard TEX assumptions.
- When dealing with PostScript (and in particular PostScript fonts), one should remember that the PostScript "point" (before any transformations have been applied to the user space) is equivalent to the TEX "big point" [4][16]. This distinction is important. When a user requests Palatino-Roman at 10pt, the driver should produce Palatino-Roman at 10 TEX points. Some fonts are special cases; *Sonata* was designed with the convention that the point size would be equal to the distance between the centerlines of the top and bottom lines of the staff character [5]. This convention should be maintained. The important thing is that the size of a "point" within a TEX file should always be consistent.

### Accessing specials from TEX

Specials should be generated by macros whenever possible. Users should not have to learn to use another user interface; they should just have a few new macros. `\landscape` could be a macro meaning that the current page should be rotated about the origin and then translated by *pageheight* − 2 *inches* in the $-h$ direction. This would simplify the user interface greatly.

Some extra macros would also simplify the `global` problem slightly. `\everybop` and `\everyeop` token-list registers could be defined which would be placed by the output routine at the beginning and ending of every page. Also, a `\globalspecial` register could be used. If this register was defined before the first `\eject`, the output routine would ensure that the contents occurred first on the very first page, before any *set* or *put* commands. This would require only trivial modification to the output routines, and the benefits would be enormous.

### Handling device-dependence

As much as we try to avoid it, some specials are going to be device-dependent. This is a hazard of using specials. The trauma could be reduced, however. METAFONT has a nice little feature in its ability to handle "modes" [14]; different modes for different printers. Such a facility could be implemented in TEX with macros. It would rarely be used, but it would help with certain device-dependent specials. Typing `\mode{X9700}` would define certain special-generating macros to fit the device. Modes would be defined in a `local.tex` file, just as METAFONT's modes are usually defined in a `local.mf` file. This would greatly simplify the process of producing proof copy on one device and then moving to another device for the final output. Note that it may be advisable to name the modes after the specific driver rather than the printer.

### Miscellaneous recommendations for driver behavior

Robert McGaffey's article, "The Ideal TEX Driver" [18], offers some excellent recommendations for the behavior of DVI drivers. Some of his suggestions (those regarding the handling of missing fonts) are also good candidates for handling with specials. We would like to offer some additional recommendations.

- The driver should be able to read a configuration file on startup, so that the user can include certain default options there. McGaffey mentions this, but the idea is important enough to merit discussion here as well. Possibly a system-wide configuration file should be read first, and then the user's. This might be one solution to the problem of global specials.
- The driver should also be able to look at certain environment variables defined by the user. Where the authors work, default paper type and destination are set in this way.
- Where there is complex logic (such as sorting logic), it is inevitable that a situation will surface where that logic is inappropriate. The driver should provide an option to turn off that complex logic and go back to the brute force method.
- In the configuration file or an environment variable, the user should be able to specify a "path" for the location of TFM and bitmap files, so that personal METAFONT-generated fonts can be used with ease. This mechanism should be able to deal with the various complicated and contradictory conventions for naming and storing such files. The authors are working on a mechanism for a driver driver which apparently will permit such specification in a very general way.
- There is naturally some confusion about how to handle the PostScript fonts and their ability to be scaled. TFM files for those fonts should probably be generated assuming a default size of 10 points, to be consistent with standard TEX assumptions.
- When dealing with PostScript (and in particular PostScript fonts), one should remember that the PostScript "point" (before any transformations have been applied to the user space) is equivalent to the TEX "big point" [4][16]. This distinction is important. When a user requests Palatino-Roman at 10pt, the driver should produce Palatino-Roman at 10 TEX points. Some fonts are special cases; *Sonata* was designed with the convention that the point size would be equal to the distance between the centerlines of the top and bottom lines of the staff character [5]. This convention should be maintained. The important thing is that the size of a "point" within a TEX file should always be consistent.

## Conclusion

While the usefulness of special commands is beyond doubt, the subject is extemely complex. If a driver attempts to provide a versatile command set, there are many difficult problems which must be resolved. With careful thought and planning, however, it seems certain that a powerful, useful standard set of specials can be devised, and hopefully with TUG's sanction it will be used in most available DVI drivers.

## Bibliography


[1] Adobe Systems Incorporated. "Adobe Systems Document Structuring Conventions, Version 2.0." Palo Alto, California, January 31, 1987. Manuscript.

[2] Adobe Systems Incorporated. *Colophon4*. May 1987.

[3] Adobe Systems Incorporated. "Encapsulated PostScript File Format, Version 2.0." Palo Alto, California, March 12, 1987. Manuscript.

[4] Adobe Systems Incorporated. *PostScript Language Reference Manual.* Reading, Massachusetts: Addison-Wesley, 1985.

[5] Adobe Systems Incorporated. "*Sonata* Technical Design Specification." Palo Alto, California, January 30, 1987. Manuscript.

[6] Batchelder, N., and T. Darrell. "Bringing troff up to Speed." *Unix Review* **5** (July 1987): 45–51.

[7] Bentley, Jon L. "Little Languages." *Communications of the ACM* **29** (August 1986): 711–721.

[8] Bentley, Jon L., and Brian W. Kernighan. "GRAP—A Language for Typesetting Graphs." *Communications of the ACM* **29** (August 1986): 782–792.

[9] Foley, J. D., and A. Van Dam. *Fundamentals of Interactive Computer Graphics.* Reading, Massachusetts: Addison-Wesley, 1982.

[10] Gourlay, John S. "A Language for Music Printing." *Communications of the ACM* **29** (May 1986): 388–401.

[11] Gourlay, John S., Allen Parrish, Dean K. Roush, F. Javier Sola, and Yiling Tien. *Computer Formatting of Music.* Technical Report OSU–CISRC–2/87–TR3, Ohio State University, 1986.

[12] Hosek, Don. "Proposed `DVI \special` command standard." Harvey Mudd College: Claremont, California, August 6, 1987. Manuscript.

[13] Kim, Scott. "The Three-Fold Way." *Publish* **2** (June 1987): 124.

[14] Knuth, Donald E. *The METAFONTbook.* Reading, Massachusetts: Addison-Wesley, 1986.

[15] Knuth, Donald E. *TEX: The Program.* Reading, Massachusetts: Addison-Wesley, 1986.

[16] Knuth, Donald E. *The TEXbook.* Reading, Massachusetts: Addison-Wesley, 1984.

[17] Knuth, Donald E., and Pierre MacKay. "Mixing right-to-left texts with left-to-right texts." *TUGboat* **8** (April 1987): 14–25.

[18] McGaffey, Robert W. "The Ideal TEX Driver." *TUGboat* **8** (July 1987): 161–163.

[19] McGilton, Henry, and Bill Tuthill. "Progress Through Accretion." *Unix Review* **5** (July 1987): 36–41.

[20] Reid, Thomas J. "TEXROX Installation and Theory of Operation," College Station, Texas. In preparation.

[21] Sproull, Robert F., and Brian K. Reid. *Introduction to Interpress.* El Segundo, California: Xerox Corporation, 1984.

# Site Reports

## TEX Information for Users in the U.K.

Graham Toal
Edinburgh, Scotland

Here is the definitive information for TUG members in the U.K. interested in TEXhax and U.K.-specific TEX info.

TEXhax is available by mailing to:

texhax-request@uk.ac.ucl.cs.nss

Issues of TEXhax will be mailed to you when they arrive.

There is a local U.K. equivalent of TEXhax, called UKTEX, to which you can subscribe by mailing to:

info-tex-request@uk.ac.aston.mail

This is organized by Peter Abbott, of the University of Aston. Aston also keep a LATEX archive, and will mail you details of FTP access when you subscribe to UKTEX.

People should check first to see if their local site already subscribes to the above, and whether or not they are available on a local bulletin-board or news service before subscribing personally. Sites with a large TEX community might consider a local redistribution of the above to save net traffic.

Another source of info is TEXline — a newsletter edited by Malcolm Clark. You can subscribe to TEXline by mailing Malcolm at

texline@uk.ac.ic.cc.vaxa

Non electronic-mail TEX users can get Malcolm's newsletter by writing to

Malcolm Clark,
Imperial College Computer Centre,
Exhibition Road,
London SW7 2BP
U.K.

or phoning

(01) 589-5111 ext 4949

Telex: 261503 IMPCOL G

Fax: (01) 584-7596

## TEXhax Notes

Malcolm Brown

Business continues to be brisk for TEXhax; as of this writing, 77 issues have been sent out in 1987. The distribution list at Score has become rather substantial with some 634 entries. Even this number does not accurately convey how many people are receiving the digests, since approximately 20 percent of the entries appear to be local redistribution points.

In the interests of keeping the Score list manageable, I have been encouraging folks who submit subscription requests to first check on local redistribution. The most important of these lists is TEX-L, which redistributes the digests for users on Bitnet. It also archives past issues and files that are too long for distribution in the digest. As in the past, I encourage all users on Bitnet to take advantage of the list server's services.

There are redistribution points for Japan, Australia and the United Kingdom. If you are in one of these countries and want to subscribe to TEXhax, the contacts for these lists are (respectively):

| | |
|---|---|
| Takagi Shigeyuki | takagi%icot.jp@relay.cs.net |
| Robert Elz | munnari!elz@uunet.uu.net |
| Irene Hassler | Postie@nss.cs.ucl.ac.uk |

Volunteers who would be interested in setting up wide-area redistribution services, especially in Europe, are invited to get in touch with me.

Below is a list of entries from the Score distribution list that appear to be redistribution points. I encourage anyone who currently receives TEXhax or would like to subscribe to check this list. The name of the person who originally set up the redistribution is given in the comment field (enclosed between the "!" and "!-"), except in those cases in which the person identified him- or herself only by an e-mail address.

```
"apollo!texhax-list"@EDDIE.MIT.EDU,           ! Rob Stanzel (MIT) !-
"harvard!adelie!TeXhax"@harvard.harvard.edu,  ! Barry Burke !-
"mcvax!olsen!texhax-dist"@seismo.CSS.GOV,     ! Bill Kelly !-
"mcvax!ukc!praxis!texhax"@seismo.CSS.GOV,     ! Tim Rylance !-
"ucsbcsl!tex-hax"@UCBVAX.BERKELEY.EDU,        ! Ed Sznyter !-
"ukma!texhaxuk"@anl-mcs.arpa,                 ! David Herron !-
ARI_TEX@ari-hq1.arpa,                         ! Elizabeth Gayman !-
BB_TEXHAX%M_DSAVX1%SLB-DOLL.csnet@RELAY.CS.NET, ! C. Scott Willy !-
bbd-texhax@NRL-AIC.ARPA,                      ! Navy Center Applied Res AI !-
bboard.texhax@R20.UTEXAS.EDU,                 ! Univ. of Texas bboard !-
BCO-TeXhax@BCO-MULTICS.ARPA,                  ! Rick Kissel !-
dist-texhax@LOUIE.UDEL.EDU,                   ! Univ of Deleware bboard !-
gate-texhax@rochester.arpa,                   ! Tim Becker !-
incoming-texhax@SAIL.STANFORD.EDU,            ! Joe Weening !-
Incoming-TeXhax@GSB-WHY.STANFORD.EDU,         ! !-
INCOMING-TEXHAX@sumex-aim.stanford.edu,       ! SUMEX bboard !-
info-tex%germany.csnet@RELAY.CS.NET,          ! Jochen Grobholz !-
info_tex%sask.BITNET@WISCVM.WISC.EDU,         ! Dave Emigh (Saskatchewan) !-
infotex%LLL.DECNET@LLL-ICDC.ARPA,             ! Mark Stewart !-
INFOTeX%UBVMSA.BITNET@forsythe.stanford.EDU,  ! bboard SUNY Buffalo !-
isi-texhax@A.ISI.EDU,                         ! Sam DelaTorre USC bboards !-
LOCAL-TEXHAX@USC-ECL.ARPA,                    ! !-
ML_TEXHAX%math.waterloo.edu@RELAY.CS.NET,     ! John Sellens !-
nu-texhax%northeastern.edu@relay.cs.net,      ! Dave Glaubman !-
Sierra-TeXhax@sierra.stanford.edu,            ! bboard at Stanford !-
tex-hax%corwin.ccs.northeastern.edu@relay.CS.NET,  ! Pedro de Rezende !-
tex-hax@rti.rti.org,                          ! Mike Shaddock !-
tex-list.lsu@RELAY.CS.NET,                    ! Louisiana Sate U !-
TeX%CRNLNS.BITNET@forsythe.stanford.edu,      ! G Wood Wilson Lab Cornell !-
tex%CS.BROWN.EDU@relay.cs.net,                ! John Bazik   !-
tex%garfield.mun.cdn%ubc.csnet@RELAY.CS.NET,  ! Vincent Gogan !-
tex%searn.bitnet@WISCVM.WISC.EDU,             ! Jacob Palme !-
TeX@CS.COLUMBIA.EDU,                          ! Columbia bboard !-
tex@HI-MULTICS.ARPA,                          ! David Bergum !-
tex@MIMSY.UMD.EDU,                            ! UMaryland College Park (bbd) !-
tex@Venus.YCC.Yale.EDU,                       ! Joel Young !-
tex_people@citi.umich.edu,                    ! !-
TeX_People@um.cc.umich.edu,                   ! UofM and TeXtset !-
texdst%BROWNVM.BITNET@forsythe.stanford.edu,  ! Peter DiCamillo Brown dist !-
TEXHAX.UMASS@RELAY.CS.NET,                    ! U of Mass bboard !-
texhax-bboard@ads.arpa,                       ! !-
texhax-dist@bu-cs.bu.EDU,                     ! Phil Budne Boston U !-
texhax-dist@sei.cmu.edu,                      ! Pat Barron !-
texhax-dist@SYSTEMS.CS.CORNELL.EDU,           ! Hal Perkins !-
texhax-incoming%tilde%ti-csl.csnet@RELAY.CS.NET,     ! Joe Ramey !-
texhax-incoming@harvard.harvard.edu,          ! Harvard dis list !-
TeXhax-L%IRLEARN.BITNET@WISCVM.WISC.EDU,      ! Michale Walsh Dublin !-
texhax-list%usl.usl.edu@RELAY.CS.NET,         ! Pat Landry !-
TexHax-local.UPenn@RELAY.CS.NET,              ! U of Pennsylvania Bboard !-
TeXhax-local%SUNYBCS.BITNET@forsythe.stanford.edu,      ! John LoVerso !-
TeXhax-local@ames-io.arpa,                    ! NASA Ames Research Center !-
texhax-local@CS.PURDUE.EDU,                   ! Dan Trinkle CS Dept Purdue !-
texhax-local@DECWRL.DEC.COM,                  ! Chris Kent !-
texhax-news@CSL.SRI.COM,                      ! Rosanna Lee !-
texhax-nic@SRI-NIC.ARPA,                      ! !-
texhax-nprdc@nprdc.arpa,                      ! Ron Stanonik !-
texhax-readers%carleton.csnet@RELAY.CS.NET,   ! Dick Brown !-
```

```
texhax-redist@renoir.Berkeley.edu,          ! Jeff McCarrell !-
texhax-yale@yale.arpa,                      ! Yale bboard !-
texhax%arbortext.uucp@umix.cc.umich.edu,    ! John Buhrer ArborText !-
texhax%bboards@atrp.media.mit.edu,          ! MIT bboard !-
texhax%bugs-bunny@MC.LCS.MIT.EDU,           !  !-
texhax%chalmers.csnet@RELAY.CS.NET,         ! Thomas Johnsson !-
texhax%ci-dandelion@EDDIE.MIT.EDU,          ! Dave Watson   !-
texhax%cs.uiowa.edu@RELAY.CS.NET,           ! Nancy Jensen !-
texhax%cs.umass.edu@RELAY.CS.NET,           ! Valerie Caro !-
texhax%dabzcd.decnet@ge-crd.arpa,           ! Andy Hazeltine !-
texhax%daisy.warwick.ac.uk@Cs.Ucl.AC.UK,    ! Rob McMahon !-
texhax%desdemona.tcpip@ge-crd.arpa,         ! Mary Beth Renze !-
TeXhax%eg.ti.com@RELAY.CS.NET,              ! Tim Harrison, Texas Instru !-
TeXHAX%LiUIDA.UUCP@SEISMO.CSS.GOV,          !   !-
texhax%okstate.csnet@RELAY.CS.NET,          !   !-
texhax%okstate@RELAY.CS.NET,                ! Mark Vasoll Oklahoma St !-
texhax%PFC-VAX.MIT.EDU@MC.LCS.MIT.EDU,      ! Mark London !-
texhax%sperry-csd.csnet@RELAY.CS.NET,       ! SperryCorp bbrdMinneapolis !-
texhax%ubc.csnet@RELAY.CS.NET,              ! Univ of British Columbia !-
texhax%ubc.csnet@CSNET-RELAY.ARPA,          ! Marilyn Hay UBrit Columbia !-
texhax%ubc.mailnet@MIT-MULTICS.ARPA,        ! Univ of British Columbia !-
texhax%umn-duluth.csnet@RELAY.CS.NET,       ! Steve Miller !-
texhax%use.uio.uninett@nta-vax.arpa,        ! B Larsen (U of Oslo) !-
texhax%vdsvax.tcpip@ge-crd.arpa,            ! Keith Decker !-
texhax@a.cs.uiuc.edu,                       ! Paul Richards !-
texhax@a.psy.cmu.edu,                       ! Mark Ohlund !-
TEXHAX@ACC.ARPA,                            ! !-
TEXHAX@BNL.ARPA,                            ! Graham Campbell !-
texhax@BRL.ARPA,                            ! Ballistic Res Lab MD !-
texhax@CEEE-SED.ARPA,                       ! Richard Mattis NBStandards !-
texhax@cs.wisc.edu,                         ! Doug Banks !-
texhax@DREA-XX.ARPA,                        ! bboard Darmouth, Nova S. !-
texhax@ee.ucla.edu,                         ! Jim Brucker UCLA EE !-
texhax@eecs.nwu.edu,                        ! Jacob Gore !-
texhax@EROS.BERKELEY.EDU,                   ! Rick Spickelmeier !-
texhax@gateway.mitre.org,                   ! William Malloy !-
texhax@ipsa.arpa,                           ! Mark Saaltink !-
TeXhax@KESTREL.ARPA,                        ! !-
texhax@kl.gba.nyu.edu,                      ! Ittai Hershman  !-
texhax@mitre-bedford.arpa,                  ! Mitre bboards !-
texhax@NRL-CSS.ARPA,                        ! !-
texhax@OSLO-VAX.ARPA,                       ! U of Oslo, Norway !-
texhax@oslo-vax.arpa,                       ! Jens Thomassen !-
TeXHAX@RICE.EDU,                            ! Rice University bboard !-
texhax@SALLY.UTEXAS.EDU,                    ! Fletcher Mattox !-
texhax@SRI-SPAM.ARPA,                       ! !-
texhax@UMD5.UMD.EDU,                        ! U of Maryland College Park !-
TeXHaX@UTAH-SCIENCE.ARPA,                   ! Nelson Beebe !-
texhaxers@csvax.CALTECH.EDU,                ! Michael Lichter !-
texhaxinfo%htikht5.BITNET@WISCVM.WISC.EDU,  ! KNoyens Tilburg UNethrlnds !-
TEXLASER%HTIKHT5.BITNET@WISCVM.WISC.EDU,    ! Teun Miijssen !-
texlist@CS.UCLA.EDU,                        ! UCLA BBoard !-
texlist@entropy.ms.washington.edu,          ! James King !-
texmtg%charon.local@athena.mit.edu,         ! Jean Diaz  !-
uci-texhax@ics.uci.edu,                     ! Richard Johnson !-
XTEXHAX%VASSAR.BITNET@WISCVM.WISC.EDU,      ! Vassar bboard !-
```

## 6th Meeting of the "TEX-Interessenten" in Germany

Joachim Lammarsch

The 6th meeting of the "TEX-Interessenten" in Germany took place in Münster on October 8th and 9th. The first day an introduction was given to new TEX users, mainly about using TEX and the use of the macro package LATEX. In the afternoon there were some presentations about new hard- and software for TEX. The next day was used for short reports and discussions. Professor Dr. O. Hermann from Heidelberg presented his work with old German letters. Dr. H. Partl, Vienna, reported on his LATEX extensions for the German language (the German style files are available at LISTSERV at DHDURZ1.bitnet). Furthermore, F. Mittelbach, Mainz, spoke about his LATEX extensions wich soon will be announced in TUGboat. A. Brüggemann-Klein, Freiburg, has developed a macro package for painting trees, named TREETEX. M. D. Middleton, Regensburg, has offered a macro package especially for secretaries (it supports the German language).

There are about 180 sites where TEX is now in operation, and between 20 and 30 more where TEX will be installed before long.

The coordinator for the "TEX-Interessenten" of Germany is Joachim Lammarsch. Bernd Schulze, who introduced TEX in Germany and did the enormous work for so many years, has left the University of Bonn. Persons who want to be added to the list of the "Deutsche TEX-Interessenten" may contact

Joachim Lammarsch
Universitätsrechenzentrum
Im Neuenheimer Feld 293
D-6900 Heidelberg, West Germany
RZ92 at DHDURZ1.BITNET

## DG Site Report

Bart Childs

We have just installed TEX 2.5. It went in as easily as the previous releases. We are hard at work on a smooth interface with DG's CEO office automation system. We are in final testing phases of PostScript, HP LaserJet Plus, and ATI drivers. These will complement the existing QMS, Imagen, and DG drivers.

## Atari ST Site Report

Klaus Guntermann
Technische Hochschule Darmstadt

Since our last report in TUGboat on our TEX implementation for the Atari ST there have been major improvements:

1. METAFONT has been ported to the ST. The implementation includes online output of created characters using "switched screen". When a character shape is displayed you may switch back and forth between the image and the text screen where METAFONT reported errors and other messages. METAFONT is halted until the user resumes operation. Conversion programs from GF to DVI, PK and PXL format are included.
2. Drivers for some popular dot matrix and laser printers have been implemented (see the output device chart in this issue). All drivers use the PK font format to save disk space. The laser printer drivers use font download to achieve high throughput (the communication line usually is the bottleneck) and their latest revision can include graphics in device dependent format.
3. Currently we are updating the preview program to allow the user to move the display window faster within a page and to switch between standard size and a compressed display (to see global formatting effects) without disk accesses. Optional margins are displayed to see where the contents will be located on a printed page.
4. We have finished a prototype "remote preview" for Atari ST's connected as terminals to a multiuser host. The program "downloads" the needed characters for the pages to be seen to the Atari. Thus no fonts need be available on a local storage medium. We are improving this prototype, so this is not yet available for distribution.

Since the Hanover Fair in Spring 1987 there are negotiations to find a distributor for the United States. Within Europe there are already connections to different countries from Spain over Austria and the Netherlands up to the United Kingdom and Norway.

For up-to-date information please contact

Kettler EDV-Consulting
P. O. Box 1345, D-8172 Lenggries
Federal Republic of Germany
phone (49) 8042 8081

# Typesetting on Personal Computers

## Running TEX on a 386-based computer: Twice as fast as an AT

Mitch Pfeffer and Alan Hoenig

For the third year in a row, at about this same time, I find myself talking about running TEX on the latest generation of personal computers. In each annual installment, the speed of the machines has doubled.

The 386-based machine I'm now using runs at 16 MHz, includes 640 K of real memory, 1 Mb of extended memory, and a 40 Mb hard disk, and costs about $3,000, including a Hercules-compatible video card, and a monochrome display. Also included are two serial ports, two parallel ports (one on-board, the other on the video card), and sockets to accept either the 287 or the 387 math-coprocessor. The unit can handle up to 7 Mb of extended memory. The system comes with a one-year warranty.

The computer is made by the Fortron Corporation. To date, I've found their machines to be unusually reliable. (Disclosure Department: Fortron Corporation, in an effort to make compatibles easy-to-use for newcomers, is one of the companies that includes a copy of my book, *The Complete Computer Companion — Buying and Using Your First Personal Computer*, with every computer they sell, and they also sell the book separately.)

If speed is not a critical issue to you, the Fortron AT-class computers are well worth a look: they're about $1,000 less than the 386 machine. They come with 384 K of extended memory, and one parallel port (the one that's on the mono-graphics video card). The version I tested was their 10 MHz/no wait-state model, using the "PAT" motherboard. This motherboard is of special interest: It's designed to fit in either their AT case, or a PC case, and it's compatible with PC components — this means that you can upgrade your PC-class computer to an AT-class machine by simply swapping motherboards. This approach is a much better solution than an accelerator card, and costs about the same.

On the 386 machine, TEXing a 5-by-8 inch page, containing some embedded mathematics, takes under 2 seconds. It takes 220 seconds to compile the WEB document `tangle.tex` — that works out to around 3 seconds per page (WEB documents are about as complex a typesetting job as you can get). On the 10 MHz/no wait-state model, it took 283 seconds to compile `tangle.tex`.

The 1 Mb of extended memory comes in handy: I use it as a RAM-disk, and copy all my commonly-used programs, including my editor (a specially-modified version of PC-Write), TEX, its `plain.fmt` and `tex.poo` files, batch files, and the previewer, into this RAM-disk. (This is done automatically by the `autoexec.bat` file, whenever I turn on the computer.) By setting a `path` to the RAM-disk, I've eliminated the load time of these programs — they're up and running the instant I give the command. (In order for TEX to find its `plain.fmt` and `tex.poo` files, you'll need a utility program that does the same thing for data files as `path` does for programs.)

If you use Personal TEX's Cordata LP-300 driver (which requires RAM-disk space) on a highly-complex document, and you use your RAM-disk as explained above, you may find you don't have enough room left for the driver to run successfully. The solution is to set up the batch file you use to activate the print program as follows:

```
del e:\tex.exe
del e:\plain.fmt
pclaser %1 -L=c -R=e
if exist e:park!@#.cor del e:park!@#.cor
copy c:\pgms\tex.exe e:\
copy c:\pgms\plain.fmt e:\
```

This deletes the two largest files off the RAM-disk ('`e:`') and runs the print program. It then tests to see if the print program's temporary file is still present on the RAM-disk (this occurs if you've interrupted the print program); if this temporary file exists, it is deleted. Finally, the batch file copies the two large files back to the RAM-disk. Note that if you hit control-C to stop printing, you must give the batch file a moment to do the copying — if you're too impatient and hit control-C again, you'll break out of the batch file.

The computer is available with two styles of keyboards: the new enhanced style, with the function keys along the top edge, or the original AT-style, with the function keys on the left side of the keyboard. Unless you'll frequently be using other computers that have the enhanced keyboard, I'd recommend the original AT-style: it's easier to swing your hand to the left to get to the function keys, than it is to reach over the keys to get to the top of the keyboard. When it comes to the feel

of a keyboard, I still haven't found anything close to the hard-contact type of keyboard mentioned in a previous column, but the Maxi-Switch keyboard used by Fortron is a definite cut-above the pure-mush feel of most clone keyboards.

The 386 computer is hardware-compatible with the AT: it will take all of the same hard disks, keyboards, power supplies, etc., as a standard AT.

You can call Fortron at 408–432–1191, to ask for the name of your nearest dealer. If your local dealer can't match the prices I've listed, you can contact me for the names of some their dealers who can.

### On choosing a hard disk

The performance of a hard disk is determined by two characteristics: its access time, and its transfer rate. The access time tells you the average time required for the hard disk to find the file you want. The transfer rate tells you how fast the information can be transferred, once it's found. The hard disk I'm using is a Seagate ST251. This is a reasonably reliable, inexpensive hard disk, with an access time of 40 milliseconds — a satisfactory, but not earth-shaking speed for an AT- or 386-class machine. TEX, however, doesn't seem to care much about the access time of your hard disk: when I substituted a Priam hard disk, with its fast 28 ms access time, TEX's performance remained the same. I found that the other programs I use were also pretty much indifferent to the Seagate's slower access time. But if you do a lot of data-base work, or use other programs that have to scavenge all over your hard disk to find information in non-sequential order, your best bet would be the more-expensive Priam drive. (Users of the Cordata LP-300 driver may also find that they can get away with using a hard disk with a fast access time, such as the Priam, instead of the required RAM-disk. In fact, even the 40 ms Seagate seems to work.)

Priam drives are available in 45 Mb, 60 Mb, and 130 Mb versions. They have a reputation for reliability (almost a contradiction-in-terms when it comes to hard disks), and I've found the company to be responsive.

If many of your applications are disk-intensive, you might want to investigate the newer hard-disk controllers. These new controllers offer faster transfer rates, but require drives compatible with their new standards: RLL and SCSI (pronounced "scuzzy").

Because TEX accesses information on the hard disk in sequential order, I now work with a reduced number of buffers. My `config.sys` file contains the line '`buffers=17`'. This leaves a bit more room for RAM-resident programs.

## Queries

### Request for Contributions to a New Publication

The increasing use of Desktop Publishing Systems (DPSs) is leading to the widespread appearance of appalling pieces of "design", perpetrated by poor software and by people without adequate training (often through no fault of their own).

I am collecting examples of these excresences both for my own use in a planned typographical design course and for publication in a sort of "Chamber of Horrors" book, if there are enough examples to make it really *bad*!

All contributions will be gratefully received and will be acknowleged in the publication (if it gets off the ground).

Please send examples to:
Peter Flynn
Computer Bureau
University College, Cork
Ireland

If they can be sent electronically, my addresses are:
Bitnet: `CBTS8001@IRUCCVAX`
HEANET: `CBTS8001@IRL.HEA.UCC.VAX1`
KOM: `"Peter Flynn UCC"@EuroKom`
BIX: `pflynn@bytecosy`

*SUPPLICO STET CEDULA*

# Macros

## What Constitutes a Well-Documented Macro?

Christina Thiele
Carleton University, Ottawa

### Background

At the recent TEX Users Group annual meeting in Seattle, a number of us began to see that we had common problems with respect to using TEX as the means of typesetting materials for publication. We work in areas where people are designing macros, and then handing them on for general use. However, unless a macro is well-documented, it can be of little use to the ordinary user who is not a programmer, or who is not particularly interested in how TEX works. The main concern of the ordinary user is: Will this macro, or series of macros, do the job I need done with this particular piece of text?

On the second day of the conference, five of us* sat down to a working lunch to try to determine what would constitute a well-documented macro: what elements of description should it contain, how should it be presented, and in what order. We came up with what could provisionally be called "guidelines": the five basic elements outlined below. With this in hand, we then had a second meeting on Wednesday, with about 18 participants from the conference attending. Some more good ideas came out, which we then added to our first list. We now think it time to put all these ideas down on paper, and open the discussion to the general TUG community. For the confirmed TEX user, this may seem like a trivial exercise. However, if you bear with us, we hope that it will become clear that with a small, but solid corner block, this exercise should result in a number of benefits for novices and experts alike.

For example, it was suggested that a macro library be built up inside TUG, but unless the macros are well-documented, how can TUG take on the responsibility for building such a library? Would TUG want to be responsible for testing the macros, fixing any bugs or anomalies, helping people implement the macro? I think not; the onus should be on the supplier of the macro or macros to ensure the usability of what is being provided.

In that case, some sort of guidelines would be most helpful: the person writing the macro would at least have some headings or sections to complete the description. The user would be able to find out as much — or as little — as required in order to use the macro. A well-documented macro would not require the user to first disassemble the definition in order to see how to use it; the description would include this information.

Sometimes, neither the macro designer nor the user writes the documention, but a third party in the affair whose specific job it is to make the macro usable for various keyboarders. The person writing documentation may or may not be a professionally trained technical writer; the immediate task at hand is to write documentation on the local use of TEX and the local macros already designed. By establishing some sort of guidelines, the writing of macro documentation is rendered much easier; there are headings and topics to write towards, sections to fill in, comments to add, and so on. In short, guidelines make it easier to put something down on paper, beyond just the macro definition(s).

One reason for using macros is to make keyboarding work easier and the results more uniform. This becomes particularly important in a production environment, such as a journal operation, where a large volume of work is done on a regular basis. In most cases, production keyboarders are not computer experts; they are expert typists. Documented macros make it possible for them to work at top speed since a good macro description states clearly how the data should be keyed. For this and other reasons, I think you'll find that the sorts of things we consider part of a well-documented macro will respond to the actual needs of the many users and designers of macros.

### The Initial Points from the Tuesday Meeting

1. What is the shape or format to be achieved? Give a visual and/or verbal description.
2. Give instructions on how to use the macro:
   i. the skeleton of the macro, without any text;
   ii. the skeleton with text filled in;
   iii. the output of that particular text and macro;
3. Supply the macro definition, heavily annotated with % comments.
4. How is the macro related to other macros? That is, if you give the macro to someone,

* 26 August, 1987: Mary Coventry (Univ. of Washington), Helen Gibson (The Wellcome Institute, London), Regina Girouard (AMS), Stephanie O'Hara (Univ. of Maryland), and myself.

make sure that it doesn't have some strings attached.

5. Where (i.e., at what points in the definition) can the definition be modified? Include some sort of naming convention, (especially if this is done in-house). And make sure to rename a redefined macro.

### Additional Comments from the Wednesday Morning Meeting

a. "What goes in and what goes out?"

b. Make sure there are *lots* of examples, not just one sample input file with its output.

c. Provide a context for the macro: where would it be used?

d. make a list of potential error messages, and their likely source

e. Should the end-users see the technical side of the description?
*Comment*: If such macro descriptions are sent via the network, it's probably a fairly knowledgeable TEX user who's picking it up: they could then filter out the technical material before passing the description on to the end user.

f. Before making a macro part of an eventual TUG library of macros, it (the macro) should meet minimum specifications (i.e., something like these guidelines) before being included in the library. This would then keep the TUG office work to a minimum, and would also increase the degree of utility of even having such a library.

g. Include information on who wrote the macro, and how they can be reached (e-mail, telephone, mailing address).

Between the two meetings, Barbara Beeton took time to write down some of her ideas on all of this as well. The following are mainly concerned with the readability and stability of macros.

a. Use indentation to indicate "level"; that is, don't write a solid paragraph-like macro.

b. Within definitions, put a `%` after a `{` or `}` at the end of a line (but not after digits or control sequences) to help control unwanted side effects from carriage returns interpreted as spaces. If there is a good reason to do otherwise, e.g., to control `\obeylines`, *annotate* it.

c. Document anything that is even vaguely "inscrutable".

d. With respect to length of macros: Keep them shorter than about 60 lines (the length of a printed page on most line printers), and arrange them in the file so that every macro will be complete on one page in a printed copy.

e. If your system permits, put in page breaks to assist with the intent of the previous item; remember, the goal is "scrutable" printout.

f. Keep a thorough change log: date, perpetrator, etc. Put the date, etc., at top of file when a new version is released.

g. Put `\endinput` at the end of the actual macros in the file, to free anything that follows the `\endinput` from the usual TEX syntax rules. This is a natural place to put user documentation and mass commentary, so that it can all be in one file, while saving the (admittedly slight) time that TEX would require to read and ignore it if presented as comments within the body of the file.

### So What Happens Now?

As mentioned above, a number of people came up with these ideas. The discussion should now go out to the general TUG community on what constitutes a well-documented macro. In addition to comments and criticisms, which we most certainly want to receive, we thought it might be worthwhile to invite TUG members to document an existing macro implementing the ideas presented above.

Please send all correspondence — suggestions, samples, criticisms — to either of the addresses below. Helen Gibson and Regina Girouard have agreed to share the task. All mail from Europe should be forwarded to Helen; all North American mail to Reggie. We will then sort through the material, and write up what we find in upcoming issues of the TUGboat.

Helen Gibson
c/o The Wellcome Institute
183 Euston Road
London NW1 2BP
Great Britain
01-387-4477

or

Regina Girouard
The American Mathematical Society
201 Charles Street
P. O. Box 6248
Providence, Rhode Island 02940 USA
(401) 272-9500 (ext. 224)

## Macros for Jill

Donald E. Knuth
Stanford University

At the TUG meeting in July, 1986, I mentioned in conversation that one of my new household duties was to write macros for my wife Jill, who had just installed TEX on her PC. Later, when Jill came to the dinner party, many people asked her for copies of the macros; and this led eventually to the idea that I should publish them in *TUGboat*. So here they are, slightly cleaned up from the way I originally wrote them.

The first task Jill assigned me was perhaps the most interesting. She had started to keep an electronic journal, and she wanted to make a nice hardcopy book. The format she had in mind was somewhat tricky because she wanted to be able to generate marginal notes in the middle of any paragraph. Furthermore, she wanted these notes to go in the left-hand margin on left-hand pages and in the right-hand margin on right-hand pages.

This task is difficult for TEX, because TEX generates paragraphs before it knows what page they will go on. Indeed, the decision about what to put on page 100 may not be made until TEX has generated a good deal of page 101.

One way to solve the problem would be to cheat, by putting the notes in both margins and masking off the undesired ones. Jill didn't like that idea very much.

A legitimate solution can be obtained by asking TEX to make two passes over the input: The first pass writes a file that tells the page numbers of each marginal note; the second pass reads this file and puts the notes into the desired margin.

The second solution isn't terribly difficult, but I decided to use a third approach, which is surprisingly simple. TEX can easily be programmed to put all the notes in the left margin, or all in the right margin. Then we simply tell TEX to output only the left-hand pages, or only the right-hand pages. With two runs, we've got everything.

The text of the marginal notes was specified in Jill's journal by using a special case of an idea that appears in Appendix E of *The TEXbook*, where a similar notation is used for index entries. Namely, `^{note}` yields 'note' in the margin and also in the paragraph; `^^{note}` yields 'note' in the margin only.

Here is the macro file `jmac.tex`:

```
% format for Jill's Journal
% sample input:
% \input jmac
% \title A New Chapter That Starts a New Page
%
% \date Umbruary 29
%
% When I woke up this morning, I decided to make this
% journal into a book, using \TeX. I like to put ^{notes}
% into the margin, so that it's easy to find things later.
% My husband^^{Don} figured out a tricky way to put these
% notes into the left margin on left-hand pages, and into
% the right margin on right-hand pages.
%
% In order to do this, he claims that it's necessary to run
% \TeX\ on the file {\it twice\/}!^^{two runs needed} One
% run gives the odd-numbered pages, the other gives
% even-numbered pages. Fortunately, this doesn't take
% much longer, because printing is the slow part.
%
% This journal contains {\it ^{no math}}.
%
% \bye
```

```
% Each run begins with a little dialog:
\newif\ifleft
\def\lefthand{l }
\message{************************ Which pages do you want (l or r)? }
\read-1 to\next         % get user's response (l or r)
\ifx\next\lefthand\lefttrue\else\leftfalse\fi
\message{OK, I'll produce only the
 \ifleft left\else right\fi-hand pages.     }
% Here are conventions for text layout
\frenchspacing          % no extra space after punctuation
\hsize=5.25in           % lines to be 5.25 inches wide
\baselineskip=14pt      % and 14 points apart
\parindent=0pt          % no paragraph indentation
\parskip=\baselineskip  % leave a blank line between paragraphs
\topskip=5\baselineskip % leave four blank lines at top of page
\vsize=40\baselineskip  % forty lines on a page
\setbox\strutbox=\hbox{\vrule height.75\baselineskip
 depth.25\baselineskip width0pt} % this is a one-line strut
\newdimen\titleoffset   \titleoffset=1.5in    % titles move into margin
\newdimen\notespace     \notespace=.375in     % space between notes, text
\newdimen\maxnote       \maxnote=2in          % maximum width of a note
\font\titlefont=cmbx10 scaled\magstep2    % font for titles at page top
\font\datefont=cmbx10 scaled\magstephalf  % font for dates in margin
\font\notefont=cmbx10                     % font for notes in margin
\font\textrm=cmr10 scaled\magstephalf     % font for normal text
\font\textit=cmti10 scaled\magstephalf    % font for emphasized text
\font\foliofont=cmbx10 scaled\magstephalf % font for page numbers
\let\rm=\textrm \let\it=\textit \rm       % all text is either \rm or \it
\textfont2=\nullfont                      % disallow math mode
% Here ^ and ^^ are changed to \mnote, visible or invisible
\newif\ifvisible
\catcode`\^=\active
\def^{\futurelet\next\testdoublehat}
\def\testdoublehat{\ifx\next^\let\next=\silentnote
 \else\visibletrue\let\next=\mnote\fi \next}
\def\silentnote^{\visiblefalse\mnote}
\ifleft                 % do the next only if assuming left margins
 \def\title#1\par{\vfill\eject\message{#1:}
  \null\vskip-4\baselineskip
  \moveleft\titleoffset\hbox{\titlefont\uppercase{#1}}
  \vskip\baselineskip}
 \def\date#1\par{\vskip\parskip
  \moveleft\notespace
   \llap{\hbox to\maxnote{\hfil\datefont#1\unskip}}
  \nobreak\vskip-\baselineskip\vskip-\parskip}
 \def\mnote#1{\strut\vadjust{\kern-\dp\strutbox
   \vtop to\dp\strutbox{\vss \baselineskip=\dp\strutbox
    \moveleft\notespace
     \llap{\hbox to\maxnote{\hfil\notefont#1}}\null}}%
  \ifvisible#1\fi}
 \hoffset=\titleoffset
```

```
\else           % do the next only if assuming right margins
 \def\title#1\par{\vfill\eject\message{#1:}
  \null\vskip-4\baselineskip
  \moveright\titleoffset\rightline{\titlefont\uppercase{#1}}
  \vskip\baselineskip}
 \def\date#1\par{\vskip\parskip
  \moveright\notespace\rightline{%
   \rlap{\hbox to\maxnote{\datefont#1\unskip\hfil}}}
  \nobreak\vskip-\baselineskip\vskip-\parskip}
 \def\mnote#1{\strut\vadjust{\kern-\dp\strutbox
   \vtop to\dp\strutbox{\vss \baselineskip=\dp\strutbox
    \moveright\notespace\rightline{%
     \rlap{\hbox to\maxnote{\notefont#1\hfil}}}\null}}%
  \ifvisible#1\fi}
\fi             % in both cases, TeX will choose the same page breaks
% We output either left-hand or right-hand pages (only)
\output{\ifleft
 \ifodd\pageno\discard\else
  \shipout\vbox{\box255 \baselineskip=30pt \hbox{\foliofont\folio}}\fi
 \else\ifodd\pageno
  \shipout\vbox{\box255 \baselineskip=30pt \rightline{\foliofont\folio}}
  \else\discard\fi\fi
 \advancepageno}
\newbox\voidbox
\def\discard{\global\setbox255=\box\voidbox}
\outer\def\bye{\vfill\eject\deadcycles=0\end}
```

The sample file at the beginning of `jmac.tex` would be output as follows, on two pages (and in two passes), if the `\vsize` is reduced to `9\baselineskip`:

# A NEW CHAPTER THAT STARTS A NEW PAGE

When I woke up this morning, I decided to make this journal into a book, using TEX. I like to put notes into the margin, so that it's easy to find things later. My husband figured out a tricky way to put these notes into the left margin on left-hand pages, and into the right margin on right-hand pages.

**Umbruary 29**
**notes**
**Don**



**two runs needed** In order to do this, he claims that it's necessary to run TEX on the file *twice*! One run gives the odd-numbered pages, the other gives even-numbered pages. Fortunately, this doesn't take much longer, because printing is the slow part.

**no math** This journal contains *no math.*

The second task was rather different. Our collection of family recipes was kept on scraps of paper, and the pieces kept crumbling and/or getting lost. Jill decided to enter the recipes into her computer so that we could print them on file cards. This way we could keep everything in order, and we could also make sets for our son and daughter to use.

Jill worked out a system of codes that she found convenient for entering the data efficiently. The main interesting thing (to me) was the way it was possible to implement these codes as "active" characters in TEX. The trick was to define the macros first, before fooling around with active characters, so that the old character meanings wouldn't get mixed up with the new ones.

Here is the file `rmac.tex`, which should be almost "self explanatory":

```
% recipe format
% sample input:
% \input rmac
% #RELISH
% >Thanksgiving Cranberry Relish
% <Wilda Bates Carter
% $3 cups
% |chill overnight
% *
% @1 pound fresh cranberries
% 2 oranges, peeled and seeded
% rind of one orange, grated
% 1 ^1/2 c sugar
% *
% !Coarsely grind cranberries and oranges. Add rind
% and sugar. Refrigerate overnight.
% =
% #BREAD
% >Cheese Crisps
% |chill at least 2 hours, bake 20--25 minutes
% %300\0 F
% $5 dozen
% *
% @1 jar sharp cheese spread (5 ounces)
% ^1/2 c butter
% ^1/4 t salt
% dash pepper
% 1 ^1/2 c flour
% *
% !Beat together cheese and butter. Stir in remaining
% ingredients. Form into two rolls, 1~^1/4 inch in
% diameter. Wrap and chill at least 2~hours. Cut into
% ^1/4-inch slices, place on ungreased cookie sheet,
% bake 20--25 minutes at 300\0~F until slightly
% darker in color.
% =
% \bye
\hsize=4.25in
\vsize=7in
\parindent=0pt
\font\classfont=cmbx10 scaled\magstep2
\font\titlefont=cmbx10 scaled\magstep2
\font\specfont=cmsl10 scaled\magstephalf  % time, temp, qty
```

```
\font\ingredfont=cmr7 scaled\magstep2
\font\normalfont=cmr10
\newdimen\specbaseline
\specbaseline=14pt  % \baselineskip between time, temp, qty
\output{\shipout\vbox{\vbox to .75in{
   \rightline{\classfont\currentclass\hskip-.25in}\vss}
  \nointerlineskip\box255}
 \advancepageno \global\let\currentdonor=\empty}
\let\currentdonor=\empty
\def\0{$^\circ$}    % degrees
\obeylines
\def\class#1
 {\gdef\currentclass{#1}}
\def\title#1
 {{\message{#1}\titlefont#1\par}}
\def\donor#1
 {\gdef\currentdonor{#1}}
\def\time#1
 {{\baselineskip=\specbaseline \rightline{\specfont#1\/}}}
\def\temp#1\0 F
 {{\baselineskip=\specbaseline \rightline{\specfont#1\/\0 F\/}}}
\let\quantity=\time
\def\ingredients{\ingredfont\everypar{\hangindent=20pt}}
\def\method{\let^^M=\space \normalfont \everypar{}}
\def\endit{\par\vfill%
 \ifx\currentdonor\empty\else\rightline{---\currentdonor}\fi%
 \eject\obeylines}
\def\frac#1/#2{\leavevmode\raise.5ex\hbox{\the\scriptfont0 #1}%
 \kern-.1em/\kern-.15em\lower.25ex\hbox{\the\scriptfont0 #2}}
\catcode`\"=14                          " comment character
\catcode`\#=\active \let#=\class        " class of food, e.g. SOUP
\catcode`\>=\active \let>=\title        " name of recipe
\catcode`\<=\active \let<=\donor        " source of recipe
\catcode`\|=\active \let|=\time         " preparation time
\catcode`\%=\active \let%=\temp         " baking temperature
\catcode`\$=\active \let$=\quantity     " amount of output
\catcode`\@=\active \let@=\ingredients  " begin list of inputs
\catcode`\!=\active \let!=\method       " begin cooking algorithm
\catcode`\*=\active \let*=\medskip      " spacer
\catcode`\^=\active \let^=\frac         " numerator of fraction
\catcode`\==\active \let=\endit         " end of recipe card
```

Notice the use of `\obeylines` here: Most of the data for a recipe appears on single lines, until you get to the "method" which consists of one or more paragraphs. Therefore `\method` converts the ends of lines to spaces. The method is followed by an '='; this finishes the card and restores `\obeylines` mode.

If the `\vsize` is reduced to 2.5 inches, the sample input produces the two cards of output shown on the next page.

Since we computerized our recipes in July, we've used the resulting cards quite often. Jill's format has worked well; it's easy to read the recipes while fixing the food, and it's easy to plan ahead because the quantities and preparation are highlighted.

Of course, the next step should be to connect the computer to our kitchen equipment, so that the cooking will be done automatically. But I think I'll work on *The Art of Computer Programming* first.

# RELISH

## Thanksgiving Cranberry Relish

*3 cups*
*chill overnight*

1 pound fresh cranberries
2 oranges, peeled and seeded
rind of one orange, grated
1 1/2 c sugar

Coarsely grind cranberries and oranges. Add rind and sugar. Refrigerate overnight.

—Wilda Bates Carter

# BREAD

## Cheese Crisps

*chill at least 2 hours, bake 20–25 minutes*
*300° F*
*5 dozen*

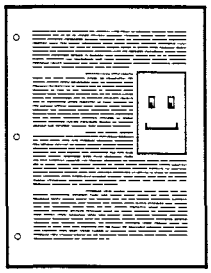
1 jar sharp cheese spread (5 ounces)
1/2 c butter
1/4 t salt
dash pepper
1 1/2 c flour

Beat together cheese and butter. Stir in remaining ingredients. Form into two rolls, 1 1/4 inch in diameter. Wrap and chill at least 2 hours. Cut into 1/4-inch slices, place on ungreased cookie sheet, bake 20–25 minutes at 300° F until slightly darker in color.

# Floating Figures at the Right — and — Some Random Text for Testing

Thomas J. Reid
Texas A&M University

PREFACE: This article is a rewritten version of a note which the author sent to *TEXhax*. The techniques presented in this article represent several significant improvements over those in the earlier note. In particular, the output routine has been simplified and generalized and the `\everypar` token list is now used to control the figure insertion.



Placing a figure to the right of a paragraph of text is relatively easy in TEX. It is done by placing the figure before the start of the paragraph and using `\hangindent` and `\hangafter` to set the amount of the indent and the number of lines needed to "cover" the figure. TEX will automatically adjust the paragraph shape to fit around the figure as it has done here.

However, in practice, the figure is likely to be larger; it is very probable that one paragraph of text may not be enough to "cover" the figure. The figure above shows a sample page of "text" containing such a figure. This smaller figure begins even with the second "paragraph" and extends into the third one. While hanging indents can be used easily to set the shape for paragraph two, their use for paragraph three is more complicated: it is necessary to account for the size of paragraph two and the `\parskip` glue when calculating the number of lines to be indented.

A further complication can arise. Suppose that the desired place to begin the figure had been at the start of paragraph four. There is not sufficient space left on the page to begin the figure; the figure needs to be floated to the next page.

## Approach to solution

This problem can be solved by performing a test at the start of each paragraph of text in the area where the figure is to be placed. At the start of each paragraph, one of three possibilities exists:

1. The figure has not yet been started;
2. The figure has been started and the shape of the paragraph about to begin is to be adjusted for the remaining portion of the figure; or
3. The text has proceeded past the end of the figure.

If the figure has not yet started, we check to see if there is room left for it on the current page. If there is, we start the figure and then adjust the paragraph shape. Otherwise, we defer further action on it until the start of the next paragraph. If the figure has been started but the text has not yet covered it (e.g., paragraph three in the sample page figure), we adjust the paragraph shape using the original height of the figure minus the height which has already been covered. Once the text has passed the figure, no action is needed for adjusting paragraph shapes.

Implementation of this procedure requires that the current position on the page be known. This information is available only within the output routine; to get it, we need to change the output routine to save it for us.

## The modified output routine

Changes are made to the output routine to allow us to use it to query page height without disturbing its normal operation. These changes consist of adding a reference to an `\outputpretest` token list to the start of the output routine and making the execution of the original output routine conditional upon a flag set by the `\outputpretest` control sequences.

```
\newif\ifoutput
\newtoks\outputpretest
\edef\oldoutput{\the\output}
\output={\the\outputpretest
  \ifoutput \oldoutput \fi}
\outputpretest={\outputtrue}
```

(And all along you thought output routines were nasty beasts.)

## Preparing the figure

The figure to be inserted is defined and placed in a box register named `\figbox`. It is important to set the dimensions of this box register to reflect the dimensions of the figure. Another dimension value is set in `\figgutter`; this represents the space to be placed between the text and the figure.

```
\newdimen\unit \unit=5pt
\def\point#1 #2 #3 {\rlap{\kern#1\unit
  \raise#2\unit \hbox{#3}}}
\newbox\figbox
\newdimen\figgutter \figgutter=1pc
```

```
\def\vr<#1,#2,#3>{\vrule height #1
  depth #2 width #3}
\setbox\figbox=\vbox to 50\unit{\hbox{%
  \point  0  0 {\vr<1.5pt,0pt,30\unit>}
  \point  0 50 {\vr<0pt,1.5pt,30\unit>}
  \point  0  0 {\vr<50\unit,0pt,1.5pt>}
  \point 30  0
    {\kern -1.5pt \vr<50\unit,0pt,1.5pt>}
  \point  5 15 {\vr<2.0pt,0pt,20\unit>}
  \point  5 15 {\vr<4\unit,0pt,2.0pt>}
  \point 25 15
    {\kern -2.0pt \vr<4\unit,0pt,2.0pt>}
  \point  7 28 {\vr<1.0pt,0pt,4\unit>}
  \point  7 35 {\vr<0pt,1.0pt,4\unit>}
  \point  7 28 {\vr<7\unit,0pt,1.0pt>}
  \point 11 28
    {\kern -1.0pt \vr<7\unit,0pt,1.0pt>}
  \point  7 28 {\vr<2.5\unit,0pt,2\unit>}
  \point 19 28 {\vr<1.0pt,0pt,4\unit>}
  \point 19 35 {\vr<0pt,1.0pt,4\unit>}
  \point 19 28 {\vr<7\unit,0pt,1.0pt>}
  \point 23 28
    {\kern -1.0pt \vr<7\unit,0pt,1.0pt>}
  \point 19 28 {\vr<2.5\unit,0pt,2\unit>}
\hss}\vss}
\wd\figbox=30\unit
```

### Controlling the figure placement

In addition to the modified output routine, we define an alternate `\outputpretest` routine which will be used to decide whether or not to actually output anything to the DVI file.

```
\newbox\pagebox
\newdimen\pageht
\newif\iftryingfig    \tryingfigfalse
\newif\ifdoingfig     \doingfigfalse
\newif\ifpageafterfig \pageafterfigfalse
\def\dofigtest{%
  \ifnum\outputpenalty=-10001
    \setbox\pagebox=\vbox{\unvbox255}%
    \global\pageht=\ht\pagebox
    \outputfalse
    \unvbox\pagebox
  \else
    \outputtrue
    \ifdoingfig
      \global\pageafterfigtrue
    \fi
  \fi}
```

In this alternate pre-test routine, we test for a penalty value of $-10{,}001$. This special penalty value will be used in a later macro to signal our intentions to the `\outputpretest` routine. If this value is found, we get the height of box 255 (`\unvboxing` it removes the glue from the bottom of the box). This height is saved so that it can be used later and the box contents are returned to the main vertical list while `\outputfalse` is set to bypass execution of the original output routine.

If any other penalty value caused the output routine to be entered, the `\dofigtest` macro sets `\outputtrue` so that the normal output routine will be performed. However, a flag (`\pageafterfigtrue`) is set if we were "actively doing the figure" when the page break occurred. This condition occurs when the figure is placed flush with the bottom of the page.

Next, the macro which controls the figure placement is defined.

```
\newdimen\startpageht
\newdimen\htdone \htdone=0pt
\edef\oldeverypar{\the\everypar}
\everypar={\tryfig \oldeverypar}
\def\tryfig{%
  \iftryingfig % ------------ Section A
    {\everypar={\relax}\setbox0=\lastbox
      \parindent=\wd0 \parskip=0pt \par
      \penalty-10001 \leavevmode}%
    \dimen0=\vsize
    \advance\dimen0 by -\pageht
    \advance\dimen0 by -2\baselineskip
    \ifdim\dimen0>\ht\figbox
      \dimen0=0.3\baselineskip
      \vrule depth \dimen0 width 0pt
      \vadjust{\kern -\dimen0
        \vtop to \dimen0{%
          \baselineskip=\dimen0
          \vss \vbox to 1ex{%
            \hbox to \hsize{\hss
              \copy\figbox}\vss}\null}}%
      \global\tryingfigfalse
      \global\doingfigtrue
      \global\startpageht=\pageht
      \global\htdone=0pt
      \dohang
    \fi
  \else % ------------------- Section B
    \ifdoingfig
      {\everypar={\relax}\setbox0=\lastbox
        \parskip=\wd0 \parskip=0pt \par
        \penalty-10001 \leavevmode}%
```

```
    \global\htdone=\pageht
    \global\advance\htdone by
                          -\startpageht
    \ifpageafterfig
      \global\doingfigfalse
    \else
      \dimen0=\ht\figbox
      \advance\dimen0 by 0.5\baselineskip
      \ifdim\htdone<\dimen0
        \dohang
      \else
        \global\doingfigfalse
      \fi
    \fi
  \else % ----------------- Section C
    \global\outputpretest={\outputtrue}%
  \fi
\fi}
```

Although this macro is fairly long, it is rather straightforward (with two exceptions). It is divided (as indicated by the comments) into three sections: section A which starts the figure; section B which controls the paragraph shapes after the figure has been started; and section C which is performed after the text has passed the figure. Execution of one of the three sections is determined by the settings of `\iftryingfig` and `\ifdoingfig`.

Within section A, we first invoke the output routine to get the current page height. This is a little bit tricky since we have just started the paragraph and are in horizontal mode (`\tryfig` is called from the `\everypar` token list immediately *after* entering horizontal mode and inserting the `\parindent` glue). Thus, we need to define a temporary "dummy" `\everypar` token list (to prevent endless recursion), then break out of the paragraph, signal the output routine, and restart the paragraph.

It is then a simple matter to compute the space left on the page and test to see if that is greater than the height of `\figbox`. (The extra two times `\baselineskip` is added to avoid a problem situation. For more details, see under "Problems with the insertion macros.") If there is not enough room, we will exit the macro without changing `\iftryingfig`. This will cause section A to be checked again at the next paragraph break. If there is room on the page for the figure, we output the figure and change the `\iftryingfig` and `\ifdoingfig` flags. The height of the page when the figure is saved since it will be needed later. Then, `\htdone` is preset to zero. This indicates how much of the figure has been covered by text and it is used in calculating the number of lines to be shortened. Finally, we call the `\dohang` macro to calculate and set the hanging indent values.

Inserting the figure represents another tricky situation. The commands here use the same techniques as those given for Exercise 14.28 in *The TEXbook*: We insert a strut in the current line to give it a known depth; then, with a `\vadjust`, we insert a box with zero height and a depth equal to that of the strut. Inside this `\vtop` box, we define a `\vbox` to force the figure to have a height equal to the x-height of the current font. Finally, the innermost `\hbox` causes the figure to be right aligned.

Section B also begins by calling the output routine to get the page height. We then compute the amount of the figure which has been covered. If a page break occurred since we started the figure (`\pageafterfigtrue`) or if the figure has been covered, we set `\doingfigfalse` and terminate the macro. Otherwise, call `\dohang` again to compute and set the hanging indents for the new paragraph.

Section C is quite simple: we redefine `\outputpretest` so that `\dofigtest` won't be called anymore.

Note that we redefine the `\everypar` token list when the `\tryfig` macro is defined. This will cause `\tryfig` to be invoked at the start of every paragraph following the definition. However, the initial settings of `\tryingfigfalse` and `\doingfigfalse` result in Section C being executed each time. Sections A and B won't be used until we activate them.

To activate the insertion process, we define a macro to perform the needed setup.

```
\def\rightinsert{%
  \outputpretest={\dofigtest}
  \tryingfigtrue \doingfigfalse
  \pageafterfigfalse}
```

Finally, the `\dohang` macro is defined which actually sets the paragraph shape.

```
\newcount\hangcount
\def\dohang{%
  \dimen0=\ht\figbox
  \advance\dimen0 by -\htdone
  \advance\dimen0 by 1.49\baselineskip
  \hangcount=\dimen0
  \divide\hangcount by \baselineskip
```

```
\dimen0=\wd\figbox
\advance\dimen0 by \figgutter
\global\hangafter=-\hangcount
\global\hangindent=-\dimen0}
```

There are two interesting points to note in the \dohang macro. First, the calculation of the height of the paragraph which is to be cut out involves adding 1.49 times \baselineskip. This increase is done to provide a minimum of one half of \baselineskip of gutter space between the bottom of the figure and the x-height of the line of text below the figure. Since the top of the figure is even with the x-height of the first line, we need to increase its height by one half of \baselineskip. The additional .99\baselineskip is added since numbers are truncated on division; we want the \divide to give us a "ceiling" result, not a "floor."

The second point of interest in \dohang is the actual calculation of \hangcount. Here, we set a count variable to a dimension and then divide the count by another dimension. TEX allows dimensions to be coerced into numbers; the number becomes the value of the dimension in units of scaled points. By dividing a coerced dimension by another dimension, the units are cancelled out leaving us with a count.

### Applying the insertion macros

Now that all the pieces have been defined, we can see how to combine them within a document to set the figure. The following TEX commands show the portion of an input file where the figure is to be used.

```
  (define macros previously shown in article)
  (text for paragraph preceding figure)
\rightinsert
\par
  (text for first paragraph after figure)
\par
  (text for next paragraph)
\par
  (text for still another paragraph)
    ...
```

When using these macros, care should be taken when using grouping. The macros have been designed to allow a group to be entered after the figure has been started. However, the \rightinsert call itself should *not* be placed within a group unless you are certain that the entire figure will be covered before the group ends.

### Demonstrating the insertion macros

To demonstrate these macros under a variety of circumstances, we need to vary the amount of text prior to starting the figure and the amount which is placed after its start. This brings us to the second major topic of this article: generating variable amounts of random text.

While this may seem like a frivolous application of TEX, it does serve some useful functions. When showing how page layouts will look, typographers often use Latin text to fill out pages. For those of us who don't have any Latin text handy or don't want to type it in, it is possible to have TEX make up some text with the aid of a pseudo-random number generator.

Another benefit of randomly-generated text is that by shifting the position within the random sequence, we can vary the amount of text that will be created. This technique will be used to show how the figure placement macros act under different conditions.

### Random numbers in TEX

To start, we need a random number generator. The following one uses the linear congruential method. It has a period of 50,000 numbers with chi-squared values for 1000 number sequences falling within the 35% to 70% range. A call to \rnd results in \rndval being set to the next number in the pseudo-random sequence. The number will be between 0 and 99, inclusive.

```
\newcount\rndnum
\newcount\rndval
\newcount\rndtemp
\rndnum=0
\def\rnd{%
  \global\multiply\rndnum by 371
  \global\advance\rndnum by 1
  \ifnum\rndnum>99999
    \rndtemp=\rndnum
    \divide\rndtemp by 100000
    \multiply\rndtemp by 100000
    \global\advance\rndnum by -\rndtemp
  \fi
  \global\rndval=\rndnum
  \global\divide\rndval by 1000 \relax}
```

Now, the random number generator can be used to generate random paragraphs consisting of a random number of sentences; random sentences made up of a variable number of words; and words made up from a number of randomly selected letters. Following the normal practice in English, we capitalize the first word of each sentence.

```
\newcount\ns \newcount\nw
\newcount\nc \newcount\np
\newcount\ASCII

\def\randompar{\rnd \ns=\rndval
  \divide\ns by 10\advance\ns by 3
  \loop \ifnum\ns>0 {\randomsent}.%
    \advance\ns by -1 \repeat}

\def\randomsent{\rnd \nw=\rndval
  \divide\nw by 7 \advance\nw by 5
  \ASCII="41
  \loop \ifnum\nw>0 { \randomword}%
    \advance\nw by -1 \repeat}

\def\randomword{\rnd \nc=\rndval
  \divide\nc by 15 \advance\nc by 2
  \loop \ifnum\nc>0 {\randomchar}%
    \advance\nc by -1 \repeat}

\def\randomchar{\rnd
  \multiply\rndval by 29
  \divide\rndval by 100
  \ifnum\rndval=26 \rndval=0 \fi
  \ifnum\rndval>26 \rndval=4 \fi
  \advance\rndval by \ASCII
  \char\rndval \global\ASCII="61}
```

### Applying the random numbers

Using the random text generator to demonstrate the figure placement macros can be accomplished with the following TEX code.

```
  (define macros for figure placement)
  (define macros for random text generation)

\message{Give me a number from 0 to 99:}
\read-1 to\mynum \ns=\mynum
\ifnum\ns>99 \ns=99 \fi

\ifnum\ns=1 Skipping 1 number.
\else Skipping \number\ns\ numbers.\fi

\loop \ifnum\ns>0 \rnd
  \advance\ns by -1 \repeat

\rnd \ifnum\rndval>49
  \parskip=6pt plus 4pt minus 2pt
  \parindent=0pt
\else
  \parskip=0pt plus 2pt
  \parindent=20pt
\fi

\def\dopar{\par}
\rnd \np=\rndval
\divide\np by 20 \advance\np by 5

\loop \ifnum\np>0 \dopar {\randompar}%
  \advance\np by -1 \repeat

\rightinsert

\par
Insert the figure here or soon after.

\rnd \np=\rndval
\divide\np by 10 \advance\np by 7

\loop \ifnum\np>0 \dopar {\randompar}%
  \advance\np by -1 \repeat

\bye
```

Those who actually try the macros presented in this article are advised to try the following sequence numbers: 7, 8, 28 and 42–43.

### Improvements to random text generator

Words composed of randomly selected letters cause several problems. First, it is a very time-consuming operation to select each character at random. Further, the words do not tend to hyphenate very well. This can result in overfull hboxes (try sequences 8, 12 or 73) or word spacing that is too loose. If one is showing a layout that has narrow columns, it is desirable for paragraphs to be set without any bad breaks.

An improvement would be to select words at random from a pre-built list of words. This requires considerably fewer random numbers. Words should be chosen of varying lengths so that the text appears to be realistic. To avoid distracting the reader into trying to make sense out of the text, the word list should be made up using a language that the reader is not likely to know. Latin perhaps?

### Problems with the insertion macros

One weakness that these macros have is that they don't account for any stretching of the `\parskip` glue. This can result in an extra line being shortened below the figure if the `\parskip` glue between the paragraphs covering the figure is allowed to stretch too much (try sequence 4).

This extra line is normally not a big problem. However, if the figure is being placed close to the bottom of the page, it is possible for the extra shortened line to be placed at the top of the next

page. The \tryfig macro avoids this problem by subtracting an extra two times \baselineskip from the remaining space on the page before testing to see if there is room.

An example of the problem can be seen by removing the extra \advance\dimen0 from Section A or \tryfig and running the macros with random sequences 42 and 43.

### Improvements to the insertion macros

When a figure won't fit on a page, it is deferred until a paragraph break on the next page. However, it does not always start with the first paragraph on the next page (try sequences 9 and 35). Sequence 55 shows a variation of this same problem. The output routine lags behind the building of the vertical list. Thus, it is possible that when this "first" paragraph break occurred, TEX was still processing the earlier page. The improvement to the figure placement would be to get the figure to be output as soon as possible after the page break.

A further refinement would be to get the figure to start at the very top of the next page. This might involve placing the figure in the middle of a paragraph.

Perhaps the definitive improvement to these figure insertion techniques would be to define a new \insert (say, \newinsert\rightins) and revise the output routine to handle a \rightins as it does other inserts. This is where output routines do become nasty beasts.

## LATEX

### Contents of LATEX Style Collection as of 6th September 1987

Ken Yap
University of Rochester

The LATEX style collection now contains the files listed below. They are available for anonymous ftp from Rochester.Arpa in directory public/latex-style. You should retrieve the file 00index first to obtain a brief description of current directory contents. The file 00directory contains a reverse time sorted list of files; this may be helpful in keeping your collection in sync with LATEX-style.

| File | Description |
|---|---|
| 00directory | |
| 00index | |
| 00readme | |
| a4.sty | Set page size to A4 |
| a4wide.sty | Adjusts width too to suit A4 |
| aaai-instructions.tex | Instructions to authors |
| aaai-named.bst | BiBTEX style to accompany aaai.sty |
| aaai.sty | Style file for AAAI conference 1987 |
| acm.bst | ACM BibTEX style |
| agugrl.sty | AGU Geophysical Research |
| *agugrl.sample | Letters style, sample |
| agujgr.sty | AGU Journal of Geophysical |
| *agujgr.sample | Research style, sample |
| amssymbols.sty | Load AMS symbol fonts |
| *article.txt | Standard files in text format, |
| *art10.txt | with places to make |
| *art11.txt | language specific |
| *art12.txt | changes indicated |
| biihead.sty | Underlined heading |
| cyrillic.sty | Load cyrillic font |
| dayofweek.tex | Macros to compute day of week and phase of moon |
| | Examples of how to use TEX arithmetic capabilities |
| deproc.sty | DECUS Proceedings style |
| deprocldc.tex | Paper that describes the above |
| docsty.c | Program to convert .doc to |
| docsty.readme | .sty by stripping comments |
| doublespace.sty | Double spacing in text |
| *draft.sty | Draft option for documents for "debugging" |
| drafthead.sty | Prints DRAFT in heading |
| dvidoc.shar1 | Sh archive of DVIDOC, DVI |
| dvidoc.shar2 | to character device filter for Unix BSD systems |
| dvidoc.sty | Style file to substitute all fonts with doc font |
| epic.shar1 | Sh archive of extended |
| epic.shar2 | picture environment |
| *espo.sty | Style file for Esperanto |
| format.sty | Print FP numbers in fixed format |
| fullpage.doc | Get more out of a page |
| fullpage.sty | |
| geophysics.sty | Geophysics journal style |
| *german.sty | Redefines keywords for German documents |

| | |
|---|---|
| ieeetr.bst | IEEE Transactions BibTEX style |
| ist21.sty | IST21 document style option for cover page |
| latex.bug | latest listing of bugs found in LATEX |
| layout.readme | Prints nice diagram |
| layout.tex | showing page parameters |
| lcustom.tex | Useful macros and definitions for LATEX |
| lfonts_ams.readme | Use AMS symbols in LATEX |
| lfonts_ams.tex | |
| lgraph.shar | Sh archive of data to graph command filter in Pascal |
| local.suppl | Supplement to local guide; describes a4, tgrind, sfwmac, trademark, lcustom, and vdm |
| *memo.sty | Memo style option |
| *mfr.sty | Modifier to memo.sty |
| *mitthesis.sty | Massachusetts Institute of Technology thesis format |
| *mitthesis.sample | |
| natsci.bst | Natural sciences generic BibTEX style |
| *natsci.sty | Formats citations created with natsci.bst |
| newalpha.bst | Modified alphabetic BibTEX style |
| *nl.sty | Style file customized for Dutch |
| nopagenumbers.doc | Remove page numbers |
| nopagenumbers.sty | |
| remark.sty | Like newtheorem but no \it |
| *resume.sty | Format for doing resumes |
| *resume.sample | Sample file |
| *rscsencode.shar | |
| sc21.sty | ISO/TC97/SC21 document style |
| sc21-wg1.sty | option for cover page |
| sfwmac.sty | Useful macros for Unix documentation |
| *showlabels.sty | Shows labels and references to them |
| siam.bib | SIAM BibTEX style |
| siam.bst | |
| siam.doc | SIAM LATEX style |
| siam.sty | |
| siam.tex | |
| siam10.doc | |
| siam10.sty | |
| siam11.sty | |
| siam12.sty | |
| slem.doc | Change \sl to \em |
| slem.sty | |
| spacecites.doc | Modified to give spacing |
| spacecites.sty | between citations |
| suthesis.doc | Stanford U thesis style |
| suthesis.sty | |
| texindex.doc | Style file and processor |
| texindex.pas | for index entries. |
| texindex.sty | Works under VMS. |
| texnames.doc | Define a couple more |
| texnames.sty | TEX names |
| tgrind.sty | Tgrind macros for LATEX instead of TEX |
| threepart.sty | Three part page headers |
| *titlepage.txt | Style file in text format to go with article.txt |
| trademark.sty | Definitions of common trademarks |
| uct10.doc | U of California thesis style |
| uct11.doc | |
| uct12.doc | |
| ucthesis.doc | |
| ucthesis.readme | |
| vdm.doc | Vienna Development Method |
| vdm.sty | LATEX style |
| vdm.tex | |
| ws87.p | Wordstar 8 bit filter |
| wsltex.c | Wordstar to LATEX filter, C version |
| wsltex.p | Wordstar to LATEX filter |
| xxxcustom.tex | Supplementary macros for xxx-tex, for some xxx |
| xxxslides.sty | Supplementary macros for SLITEX, includes slides.sty |

New entries since the last TUGboat listing are marked with an *. More submissions are very welcome. Send them to

Ken
LaTeX-Style@Rochester.Arpa
LaTeX-Style@cs.rochester.edu
..!rochester!latex-style

Editor's note: People sending future submissions should note that some gateways to Bitnet strip off everything beyond 80 columns, and perhaps corrupt some other data as well (ASCII tabs may or may not remain intact). Please structure your file so that it will survive.

### For Internet users: how to ftp

An example session is shown below. Disclaimer: ftp syntax varies from host to host. Your syntax may be different. The syntax presented here is that of Unix ftp. Comments in parentheses.

### Non-Internet users: how to retrieve by mail

An archive server for LaTeX files has been installed. Send a piece of mail to LaTeX-Style (@cs.rochester.edu, via UUCP or your favourite gateway) in the following format.

- Subject line should contain the phrase "@file request".
- The body of the mail should start with a line containing only an @ (at) sign.

**Important!** The first line following the "at" line should be a mail address **from** Rochester **to** you. (Undeliverable mail will be silently dropped on the floor.)

- Follow your return address by the names of the files you want, either one to each line, or many to each line, separated by spaces.
- End with a line containing only an @ sign.
- Case is not significant.

For example, if you are user at site.bitnet, this is what you should send: (*don't forget your address*!)

```
To: latex-style@cs.rochester.edu
Subject: @file request

@
user%site.bitnet@wiscvm.wisc.edu
00readme
00index
@
```

A word to the wise: it is best to fully qualify your mail address. Our mailer is pretty ignorant of Bitnet, CSnet or UUCP addresses unless they are in registered domains. It is best that you supply explicit gateway routes. Also use the new domainized form or addresses whenever possible because the old .ARPA addresses are fading away. Examples:

```
user%site.bitnet@wiscvm.wisc.edu
user%site.csnet@relay.cs.net
site!user@uunet.uu.net
```

Note that csnet-relay.arpa doesn't work any more. Long UUCP paths are discouraged. System administrators get upset and your turnaround is very slow anyway.

If the Subject: line looks like:

```
Subject: @file request uuencode
```

or

```
Subject: @file request rscsencode
```

then the mail will be encoded with the requested scheme before sending. This *might* help sites that get mail through gateways with unfriendly EBCDIC/ASCII mappings. You can find sources for the two types of en/decoders in the collection. You may have to do some porting of sources.

Do not include any messages in the mail. It will not be seen by human eyes. Be patient as the server is actually a batch program run once a day. Files will be sent in batches, each not exceeding 100kbytes in size.

### IBM PC and clone users: how to get a distribution

David Hopper of Toronto, Canada, is offering copies of the style collection on diskettes. This is not a commercial enterprise. David is doing this in his own time as a favour to the TeX community. The entire set of style files, as of September 15th, fits on

### Sample FTP session for Internet users

```
% ftp cayuga.cs.rochester.edu   (a.k.a. cs.rochester.edu, a.k.a. 192.5.53.209)
 ...                            (general blurb)
user: anonymous
password: <any non-null string>
ftp> cd public/latex-style      (where the files are)
ftp> ls                         (to see what is there)
 ...                            (lots of output)
ftp> get 00index
 ...                            (more blurb)
ftp> quit
```

one 1.2 MB diskette or three 360KB diskettes. No subsetting, please. Send David

1. Formatted diskettes,
2. Indication of the format required,
3. A self-addressed mailer, and
4. A $5.00 donation per set of files, to cover postage and equipment wear & tear. (If you live outside North America, airmail delivery will probably require more postage. You should probably contact David for details.)

*NB*: 1. David does not have the CTEX files. 2. No phone calls or personal visits.

David's address:

David W. Hopper
446 Main Street
Toronto, Ontario
Canada M4C 4Y2
Thanks, David.

Editor's note: Traffic on the network servers and gateways has been very high recently, and in order to provide improved service, there have been some volunteers to maintain local "slave" repositories of the LaTeX style collection. There is usually a geographic or network restriction requested, since the idea is to cut down traffic, not add to it. The following areas will be covered by the volunteers listed.

- Bitnet users: Texas A&M maintains a list- and file-server which is already handling (with TEX-L) much of the Bitnet distribution of TeXhax. An inquiry via listserv will retrieve a list of all TeX-related files:
  `tell listserv at tamvm1 get tex filelist`
- United Kingdom, for users of JANET or uucp: Stephen Page, `sdpage@uk.ac.ox.prg` or `...!ukc!ox-prg!sdpage`
- European users of BITnet: Christoph Gatzka, `zrgc002@dtuzdv5a.Bitnet`

Additional volunteers should contact Ken.

## The LaTeX User's Column

Jackie Damrau
University of New Mexico

Since there has not been much activity in this column, we have decided to try to take problems and answers from TeXhax. If anyone who submits material to TeXhax does not want an item republished, please let me know. The questions section below starts off with a few items from other mail networks.

At the TeX conference in August, there was some discussion that this column should also be a question and answer column for beginners. So **BEGINNERS**, please submit your questions to

Jackie Damrau
Dept. of Mathematics and Statistics
University of New Mexico
Albuquerque, New Mexico 87131
(505) 277-4623

or

UUCP: `damrau@ariel.unm.edu`
Bitnet: `damrau@unmb.Bitnet`

## A LaTeX Bug

We did receive a copy of a letter to Leslie Lamport about a possible LaTeX bug:

I recently uncovered a bug in LaTeX Version 2.09 (as of April 19, 1986). While inside the `tabbing` environment, every time a `\>` command appears two words on TeX's save stack get used. Thus any large `tabbing` environment can cause a save stack overflow, resulting in this error message

```
TeX capacity exceeded, sorry.
```

This is especially true if your system is configured with a small save stack size. The problem stems from your doing both global and local assignments to box register `\@curline` and is explained in *The TeXbook* on page 301.

I fixed the problem by changing this declaration

```
\def\@addfield{\global\setbox\@curline
        \hbox{\unhbox\@curline
              \unhbox\@curfield}}
```

to

```
\def\@addfield{\setbox\@curline
        \hbox{\unhbox\@curline
              \unhbox\@curfield}}
```

Note the absence of the `\global` in the redefinition. The fix seems to work fine, but I don't know if it will cause any bad repercussions somewhere else.

I also caused LaTeX's output routine to go into an infinite loop, but I hesitate to call it a bug because I have only known it to occur with one of my document styles. The problem occurred when `\output` was being called just after a `\begin{enumerate}`. Inside the output routine, TeX was executing `\par` over and over again and not making any progress. With a little hunting, I found that LaTeX disables `\par` just after the beginning of list constructions. I also narrowed down the point where the infinite loop was occurring to this construction within the header text

```
\vbox to.4in{\vfil\hsize 4in ...
                        some text ... \vfil}
```

When the second `\vfil` command got executed, TeX, for some reason, tried to execute `\par`. Since `\par` at this point was disabled, TeX kept trying.

My solution was to add this

```
\let\par=\@@par
```

to the beginning of the `\output` definition. Since, I believe, your document styles won't cause a similar construction to occur in the output routine, this change is not needed for standard LaTeX, but I believe it can help to make LaTeX more bullet-proof for users who can't leave well enough alone.

Many thanks for all your work in developing LaTeX. You have made my job of helping our staff produce TeX-quality documents immeasurably easier.

Benjamin J. Kennedy
Xybion Corporation

### Questions from the mail networks

Here are some questions, answers and macros that appeared on the mail network. If anyone has answers to these questions, please send the answers to both the originator and myself.

**Question 1.** A problem I had with verbatim was to include a file inside a `verbatim` environment. I wanted to have a file of sample LaTeX code that could be LaTeXed alone correctly and then included verbatim as an appendix to another document. Of course, I could have copied the file in with my editor, but I wanted changes in the sample file to be automatically changed in the version in the appendix.

The problem, of course, is that

```
\begin{verbatim}
\input examplefile
\end{verbatim}
```

only prints `\input examplefile`. One can place the verbatim commands in the example file, but then one can not use the example file directly to generate the example. A solution:

```
% necessary to use expandafter
\def\bv{\begin{verbatim}}
\def\infil{\input examplefile}
\expandafter\bv\infil
\end{verbatim}
```

There may be a similar solution with more `\expandafter`s that avoids the `\def`s, but at least this worked.

Terry L. Anderson
AT&T Bell Laboratories – Liberty Corners
UUCP: ...!ihnp4!kaiser!tla

Editor's note: This solution can be extended to process multiple examples in the same document in the following manner.

```
\def\infil{\input\examplefile}
\def\bv{\begin{verbatim}}
\def\doexample#1{\def\examplefile{#1}%
        \expandafter\bv\infil}
\doexample{file1}
\end{verbatim}
\doexample{file2}
\end{verbatim}
```

As Terry Anderson said above, there may be a solution with more `\expandafter`s that would avoid having to say `\end{verbatim}` explicitly for each file, but, ...

**Question 2.** I'm just a beginner with LaTeX, and I would like to define an environment of my own, including some verbatim. Something like this:

```
\newenvironment{myenv}
    {\begin{verbatim}}{\end{verbatim}}
\documentstyle{book}
\pagestyle{headings}
\begin{document}

\chapter{Hi}

Here it comes:
\begin{myenv}
Try and try
            ... again!
\end{myenv}

This is the end.
\end{document}
```

But it doesn't work. This is the most important part of the log file:

```
This is TeX, Version 2.0 for Berkeley UNIX
  (preloaded format=lplain 86.12.16)  23 Aug 87
(es.tex
LaTeX version 2.09 - Released 27 October 1986
(/usr/lib/tex/macros/book.sty
Document style 'book'. Released 4 September 1986
(/usr/lib/tex/macros/bk10.sty)
) (es.aux)
Chapter 1.
)
Runaway argument?

^^MTry and try^^M   ... again!^^M\end{myenv}^^MT\ETC
! File ended while scanning use of \@xverbatim.
<inserted text>
                \par
<*> es.tex

?
```

What happened?

Gérald Masini
Centre de Recherche en Informatique
de Nancy
uucp: masini@crin.crin.fr
CRIN, B.P. 239
54506 Vandoeuvre-les-Nancy Cedex
France

Editor's note: You get the same complaint about `\@xverbatim` when you include `\end{verbatim}` in the `\doexample` expansion shown in the note following Question 1. These problems are related. This doesn't explain why, though, and LaTeXperts are invited to provide an explanation scrutable to beginners.

**Question 3.** Tom Hoffman wrote a LaTeX macro that almost solves the `.ip` paragraph problem. [Editor's note: `.ip` is a `troff` macro indicating a labeled paragraph.] I found his mixture of TeX and LaTeX confusing, so I rewrote his solution in LaTeX:

```
\newenvironment{itemlist}[1]
  {\begin{list}{}
    {\settowidth{\labelwidth}{#1}
     \setlength{\leftmargin}{\labelwidth}
     \addtolength{\leftmargin}{\itemindent}
     \addtolength{\leftmargin}{\labelsep}
     \renewcommand{\makelabel}[1]
          {##1\hfill}}}%
  {\end{list}}
```

I also took a whack at a complete solution.

```
% Back to TeX we go.
\def\ml#1{%
        \newbox \newlabel
  \setbox \newlabel =\hbox{#1}%
      \ifdim \wd\newlabel >\labelwidth
          \newdimen \pad \pad=\textwidth
          \advance \pad by-\rightmargin
          \advance \pad by-\itemindent
          \advance \pad by-\labelsep
          \hbox to\pad {#1\hfill}%
      \else #1\hfill \fi}

\newenvironment{itemlist}[1]
  {\begin{list}{}
    {\settowidth{\labelwidth}{#1}
     \setlength{\leftmargin}{\labelwidth}
     \addtolength{\leftmargin}{\itemindent}
     \addtolength{\leftmargin}{\labelsep}
     \let\makelabel=\ml}}%
  {\end{list}}
```

This seems to work; it even nests, but the inner `itemlists` cause overfull `\hbox` errors. When I find I'm writing macros like `\ml` I usually stop, calm down, and look for another solution. However, I'm going to cut my losses and let this one go. Here's a hint for anybody who picks it up: white-space is very important (which is partly why I'm letting it go).

clayton@thumper.UUCP

**Question 4.** I am trying to use `\vfill` within the LaTeX `table` environment without much success. What I want to be able to do is to move the caption down to the bottom of the page using the rubber length `\vfill`. LaTeX seems to ignore the `\vfill` command. Am I missing something? Is there another way of doing the same thing?

My LaTeX file looks something like this:

```
\documentstyle[12pt]{report}
\begin{document}
\begin{table}[p]
\centerline{The actual table will go here}
\vfill
\caption[Test table]{This is the caption
 for the test table. It will contain
 a description of the salient features
 of the table.}
\end{table}
\end{document}
```

The `\vfill` command doesn't do anything. The caption stays 10pt below the table (or in this case the `\centerline` stuff). `REP12.STY` defines a `vskip`

of 10pt in its definition of `@makecaption` and I couldn't figure out how to change it to a rubber length. I also tried putting `\caption` in a `parbox` but that did the same thing.

Atul Kacker
...seismo!rochester!ur-tut!akk2
or akk2@tut.cc.rochester.edu

### Become a contributor!

Send in your answers to these questions. And if you create anything else you find useful, please let others share in your new found knowledge.

## Queries

### Title formatting macro wanted

I have a peculiar problenm that seems made for TEX's typesetting capabilities. The problem is the following:

Given an input list consisting of words separated by spaces (not control sequences), produce a title block of entirely capitalized words, double-spaced, centered, and (gasp!) in inverted pyramid form. I would like the lines produced in general not to be of vastly differing lengths (no long lines followed by singletons, please), and I would prefer that line breaks in titles, if the rest of the criteria are met, occur at commas in preference to just being between words.

I admit that this may be a very tall order, but as a newcomer to TEX I have quickly come to realize the vast power of this typesetting system and can only hope that such a macro already exists. If I have placed too many limitations on the macro as it stands, please inform me of any solutions to the "inverted pyramid" problem and I will fiddle with them as best as possible to produce what I need.

I am using LATEX, if that is of any help in finding an "inverted pyramid" macro.

Stephen C. Lipp
University of Texas at Austin

### Reply: Printing Out Selected Pages

Donald Knuth's reply to Helen Horstman in TUGboat Vol. 8, No. 2 (p. 217, "Print Out Selected Pages") contains a bug which can cause the omission of pages from the output, even if they were specified in the `pages.tex` file. Specifically, if the page number of the first page processed is greater than the first requested page, then no pages will be printed.

Here is an example:

```
\count0=2
This is the text for page 2.
\vfil\eject
This is the text for page 3.
\bye
```

If the `pages.tex` file contains the lines

```
1
3
```

the first page number will never be matched, and no pages will be printed.

The correction is a change to the logic in the `\shipout` macro, given below (which includes the change to the macro `\loop` given on p. 184 of the same issue). Each time TEX is to ship out a page, it tests to see if the page number is greater than the requested page. If so, it gets another number from the `pages.tex` file, continuing this test until either (a) the end of `pages.tex` is encountered, or (b) the requested page number is greater than or equal to the current page number.

```
\def\loop#1\repeat{\def\iterate
    {#1\expandafter\iterate\fi}%
    \iterate}
\def\breakout{\let\iterate\relax}
%   This lets us "escape" from a \loop
\let\repeat\fi
\def\shipout{\begingroup
    \loop
      \ifeof\pages \aftergroup\Shipout
      \else \ifnum\pageno>\nextpage
              \getnextpage
          \else \breakout
            \ifnum\pageno=\nextpage
              \aftergroup\Shipout
            \else \aftergroup\Tosspage \fi
        \fi \repeat\endgroup }
```

(Note to users of the new `\loop`: I used `\aftergroup` to "hold off" on executing either `\Shipout` or `\Tosspage` because the `\expandafter` technique doesn't quite work due to the nesting of `\if...\fi` within `\loop...\repeat`. The alternative,

```
\expandafter\Tosspage\expandafter ...
```

where we add enough `\expandafter`'s to get past the `\repeat`, is probably not worth the effort.

If a solution without grouping is required, we could take advantage of the fact that `\loop` ends with an assignment, and substitute `\afterassignment`.)

As with the original, the page numbers in `pages.tex` must appear in the order they will be generated. However, the change means that the signal value used to inform TEX that no more pages are to be printed must not be less than any page number in the file. Given a page number *larger* than any existing page number, the macro will behave correctly.

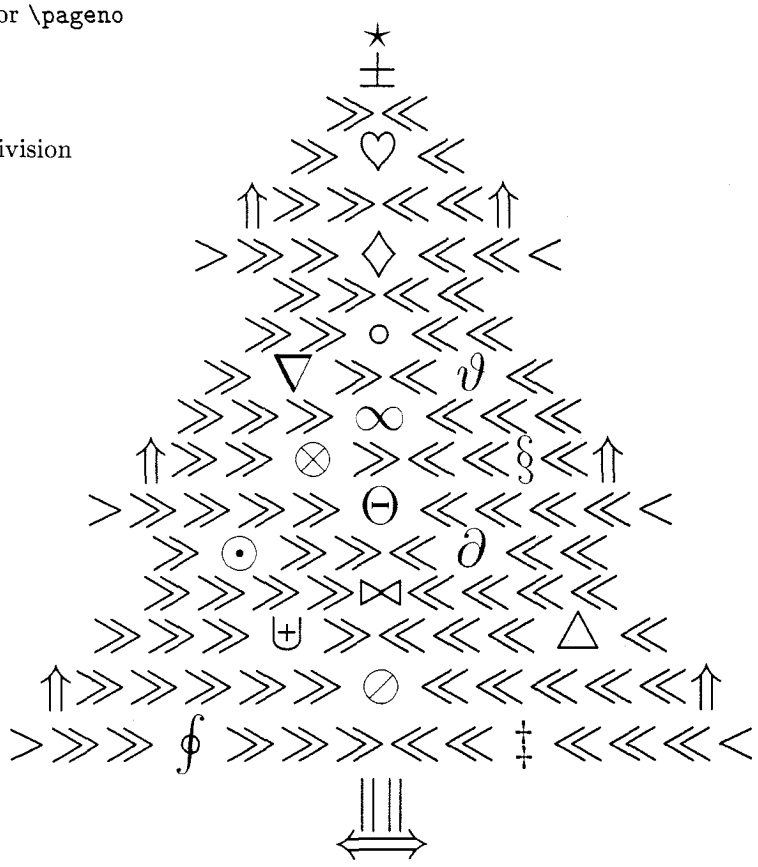
E.g., the use of "−999999" as in Knuth's example `pages.tex` file would cause `\getnextpage` to attempt to read another page number from the file, which would cause an 'eof', and all remaining pages in the file would be printed. Changing the last value to "999999" restores the intended behavior.

By the way, for LATEX users, the only change required is to substitute `\value{page}` for `\pageno` wherever it occurs.

Jeffery Boes
Lear Siegler, Inc.
Instrument and Avionic Systems Division

Editor's note: When answering a query, please send a copy of your answer to the TUGboat editor as well as to the author of the query. Answers will be published in the next issue following their receipt.

# Merry TEXmas



Editor's note: This traditional X-mas tree for mathematicians and TEXnicians was created by Irene Hyna of the Technical University of Vienna, and forwarded by Hubert Partl to TEXhax (1986, issue 15). Our warmest greetings to them both.

# Calendar

**1988**

Jan 5–9 Joint Mathematics Meeting, Atlanta, Ga. TEX Short Course, Tuesday, Jan. 5

Jan 15 **1988 TUG Annual Meeting**
Deadline for notifying program coordinator of intention to present a paper.

Jan 19 **TUGboat Volume 9, No. 1:**
Deadline for receipt of manuscripts.

Mar 18 **1988 TUG Annual Meeting**
Abstracts due to program coordinator.

Mar 21–24 RIAO 88: Conference on User-Oriented Content-Based Text and Image Handling; Massachusetts Institute of Technology, Cambridge, Mass. For information, contact Karen Daifuku, (202) 944-6252 in the U.S.; elsewhere, call the Secrétariat Général du CID in France, (1) 42 85 04 75.

Apr 4 **1988 TUG Annual Meeting**
Selection of papers, notification sent to speakers.

Apr 20–22 International Conference on Electronic Publishing, Document Manipulation and Typography, Nice, France (see announcement, TUGboat Vol. 8, No. 1, page 78)

May 16 **TUGboat Volume 9, No. 2:**
Deadline for receipt of manuscripts (tentative).

Jun 6–8 Expert Communication 88: Artificial Intelligence in Electronic Publishing; San Jose, Calif. For information, contact Marion Elledge, Graphic Communications Association, Arlington, Va., (703) 841-8160

Jun 20 **1988 TUG Annual Meeting**
Camera copy due for papers to appear in Proceedings.

Jul 18–20 TEX88 Conference, University of Exeter, England. To be placed on the mailing list, contact Cathy Booth (Janet: `booth.cm@uk.ac.exeter`) or Malcolm Clark (Janet: `texline@uk.ac.ic.cc.vaxa`)

Aug 1–5 ACM SIGGRAPH; Atlanta, Ga. For information, call (312) 644-6610

**TEX Users Group 1988 Conference**
**McGill University, Montréal, Québec**

Aug 22–24 **TUG Annual Meeting**
See call for papers, cover 3.

* * * * *

Sep 19 **TUGboat Volume 9, No. 3:**
Deadline for receipt of manuscripts (tentative).

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

*Status as of 25 October 1987*

*First Circular* *August 1987*

## TEXeter

TEX88, the Third European TEX Conference will take place at Exeter University, UK, from Monday, July 18th to Wednesday, July 20th, 1988. This conference seeks to maintain the high standard set by the previous European TEX conferences. In view of the rate at which TEX is expanding its areas of application, and the developments which will have taken place by next summer, papers are welcome from all areas of TEX, METAFONT and related subjects. Likely themes might include:

- ⋆ desktop and traditional publishing with TEX
- ⋆ document structure — LATEX, SGML, ODA, etc.
- ⋆ non-technical TEX and METAFONT applications (humanities, music, exotic languages)
- ⋆ other technical areas (chemistry, physics, biology)
- ⋆ parochial peculiarities (European considerations!)
- ⋆ ai and expert system approaches
- ⋆ TEX and *wysiwyg*
- ⋆ METAFONT and fonts
- ⋆ page description languages and `dvi`
- ⋆ micro inputs
- ⋆ macro inputs
- ⋆ standards:
  - ⋆ for `\special`
  - ⋆ for printer drivers
  - ⋆ for TEX macros
- ⋆ TEX environments

The programme will be structured to include both long and short contributions. The proceedings will be published after the conference.

### Added Value

We hope to run various workshops and participatory seminars before, during and after the conference. Specifically, we plan to organise workshops on 'the design of documents', LATEX style files (and how to modify them), and on the use of METAFONT. Seminar rooms will be available during the conference for 'birds of a feather' and other *ad hoc* group meetings. There will be facilities for exhibitions by the vendors of TEX-related software, hardware and paperware. A number of micros will be available to allow the interchange of micro software, as well as demonstrations of particular packages etc.

### What does it cost?

The total cost of the conference, including accommodation (from the evening of Sunday 17th July, through to Wednesday 20th) at Lopes Hall of Residence, meals, social activities, and the conference proceedings will be around £150. For non-residents, the charge will be about £100. The workshops and seminars before or after the conference will cost about £50 per day, which also includes accommodation and meals. Cheap rail fares within the UK are being arranged.

*First Circular* *August 1987*

**Where is Exeter University?**

Exeter is located in the sunny south west of England, close to Dartmoor, and only 10 miles from the sea. The University is approximately one mile north of Exeter city centre. There is an excellent train service to London and the North from St Davids Station. The M5 and M4 motorways provide quick and relatively painless access to London, the Midlands and the North. There is ample car parking space on campus. The local airport has daily connections to the continent, as well as UK internal destinations.

---

**Fill This in Now!**

**Name:**
**Affiliation:**
**Address:**

**Telephone:**
**email:**

Please tick where appropriate:

☐ Please add my name to the TEX88mailing list

☐ I would like to present a paper
The title will be:

---

Send to:

Malcolm Clark, TEX88
Imperial College Computer Centre
Exhibition Road
London SW7 2BP
Janet:texline@uk.ac.ic.cc.vaxa

---

TEX88 is pleased to acknowledge the support of ArborText, Addison-Wesley Publishers Ltd, Ellis Horwood Ltd (Publishers), *TEXpert systems* and UniTEX Systems.

# Institutional Members

Addison-Wesley Publishing Company, *Reading, Massachusetts*

The Aerospace Corporation, *El Segundo, California*

American Mathematical Society, *Providence, Rhode Island*

ArborText, Inc., *Ann Arbor, Michigan*

ASCII Corporation, *Tokyo, Japan*

Aston University, *Birmingham, England*

California Institute of Technology, *Pasadena, California*

Calvin College, *Grand Rapids, Michigan*

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

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

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

College of William & Mary, *Williamsburg, Virginia*

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

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

Data General Corporation, *Westboro, Massachusetts*

DECUS, L&T Special Interest Group, *Marlboro, Massachusetts*

Department of National Defence, *Ottawa, Ontario, Canada*

Digital Equipment Corporation, *Nashua, New Hampshire*

dit Company, Ltd., *Tokyo, Japan*

Edinboro University of Pennsylvania, *Edinboro, Pennsylvania*

Electricité de France, *Clamart, France*

Environmental Research Institute of Michigan, *Ann Arbor, Michigan*

European Southern Observatory, *Garching bei München, Federal Republic of Germany*

Ford Aerospace & Communications Corporation, *Palo Alto, California*

Försvarets Materielverk, *Stockholm, Sweden*

General Motors Research Laboratories, *Warren, Michigan*

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

Grinnell College, *Grinnell, Iowa*

Grumman Corporation, *Bethpage, New York*

GTE Laboratories, *Waltham, Massachusetts*

Hart Information Systems, *Austin, Texas*

Hartford Graduate Center, *Hartford, Connecticut*

Harvard University, *Cambridge, Massachusetts*

Hewlett-Packard Co., *Boise, Idaho*

Hobart & William Smith Colleges, *Geneva, New York*

Humboldt State University, *Arcata, California*

Hutchinson Community College, *Hutchinson, Kansas*

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

Illinois Institute of Technology, *Chicago, Illinois*

Imagen, *Santa Clara, California*

Institute for Advanced Study, *Princeton, New Jersey*

Institute for Defense Analyses, *Princeton, New Jersey*

Intergraph Corporation, *Huntsville, Alabama*

Intevep S. A., *Caracas, Venezuela*

Iowa State University, *Ames, Iowa*

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

Kuwait Institute for Scientific Research, *Safat, Kuwait*

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

Louisiana State University, *Baton Rouge, Louisiana*

Marquette University, *Milwaukee, Wisconsin*

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

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

Max Planck Institute Stuttgart, *Stuttgart, Federal Republic of Germany*

McGill University, *Montreal, Quebec, Canada*

National Institutes of Health, *Bethesda, Maryland*

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

New Jersey Institute of Technology, *Newark, New Jersey*

New York University, *New York, New York*

Northeastern University, *Boston, Massachusetts*

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

Pennsylvania State University, *University Park, Pennsylvania*

Personal TEX, Incorporated, *Mill Valley, California*

Purdue University, *West Lafayette, Indiana*

QMS, Inc, *Mobile, Alabama*

Queens College, *Flushing, New York*

Research Triangle Institute, *Research Triangle Park, North Carolina*

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

Ruhr Universität Bochum, *Bochum, Federal Republic of Germany*

Rutgers University, Hill Center, *Piscataway, New Jersey*

St. Albans School, *Mount St. Alban, Washington, D.C.*

Sandia National Laboratories, *Albuquerque, New Mexico*

SAS Institute, *Cary, North Carolina*

Schlumberger Well Services, *Houston, Texas*

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

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

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

Software Research Associates, *Tokyo, Japan*

Sony Corporation, *Atsugi, Japan*

Space Telescope Science Institute, *Baltimore, Maryland*

Springer-Verlag, *Heidelberg, Federal Republic of Germany*

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

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

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

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

Stratus Computer, Inc., *Marlboro, Massachusetts*

Syracuse University, *Syracuse, New York*

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

Texas A & M University, Computing Services Center, *College Station, Texas*

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

TRW, Inc., *Redondo Beach, California*

Tufts University, *Medford, Massachusetts*

TV Guide, *Radnor, Pennsylvania*

UNI.C, Danmarks EDB-Center, *Aarhus, Denmark*

University of Alabama, *Tuscaloosa, Alabama*

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

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

University of Calgary, *Calgary, Alberta, Canada*

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

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

University of California, San Diego, *La Jolla, California*

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

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

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

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

University of Delaware, *Newark, Delaware*

University of Glasgow, *Glasgow, Scotland*

University of Groningen, *Groningen, The Netherlands*

University of Illinois at Chicago, *Chicago, Illinois*

University of Kansas, *Lawrence, Kansas*

University of Maryland, *College Park, Maryland*

University of Massachusetts, *Amherst, Massachusetts*

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

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

University of Ottawa, *Ottawa, Ontario, Canada*

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

University of Tennessee at Knoxville, *Knoxville, Tennessee*

University of Texas at Austin, *Austin, Texas*

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

University of Washington, *Seattle, Washington*

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

University of Wisconsin, *Madison, Wisconsin*

Vanderbilt University, *Nashville, Tennessee*

Vereinigte Aluminium-Werke AG, *Bonn, Federal Republic of Germany*

Villanova University, *Villanova, Pennsylvania*

Washington State University, *Pullman, Washington*

Widener University, *Chester, Pennsylvania*

John Wiley & Sons, Incorporated, *New York, New York*

Worcester Polytechnic Institute, *Worcester, Massachusetts*

Yale University, *New Haven, Connecticut*

## 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 |
|---|---|---|
| | 1988 TUGboat Subscription/TUG Membership (Jan.–Dec.) – **North America**<br>New (first-time): [ ] $30.00 each<br>Renewal: [ ] $40.00; [ ] $30.00 – reduced rate if renewed before February 1, 1988 | |
| | 1988 TUGboat Subscription/TUG Membership (Jan.–Dec.) – **Outside North America**<br>New (first-time): [ ] $40.00 each<br>Renewal: [ ] $45.00; [ ] $40.00 – reduced rate if renewed before February 1, 1988 | |
| | TUGboat back issues, $15.00 per issue (19), 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; 1986 (v. 7) #1,2,3; 1987 (v. 8) #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?

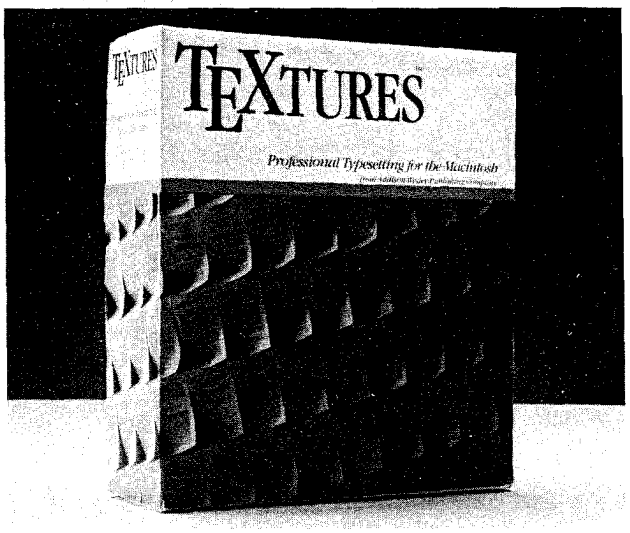
G. What are its drawbacks and how have you dealt with them?

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.

# Special Limited Time Offer to TUG Members!

## 15% off Textures




## Professional Typesetting for the Apple Macintosh

Textures is now available – the definitive TEX implementation for the Macintosh. This offer, which is available only to TUG members, ends 12/31/87.

To receive your copy of Textures, send in the coupon below. If you would like further information about Textures or about MicroTEX, professional typesetting for IBM PCs and compatibles, please call 617-944-6795.

---

Please send me _____ copies of Textures (#17220). Each package includes complete software and documentation for the special price of $420 which is 15% off list. If I am not completely satisfied, I can return the package within 60 days for full credit.

Name _____

Title _____

Company/School _____

Address _____

City _____ State _____ Zip _____

Telephone (  ) _____

Mail this coupon to Textures, EMSD Marketing, Addison-Wesley Publishing Company, 12 Jacob Way, Reading, MA 01867.

☐ My check or money order is enclosed with this order form. ☐ P.O. number _____

☐ Charge my order to my credit card ☐ Visa ☐ MasterCard ☐ American Express

Card Number _____ Expiration Date _____ Signature _____

# Introducing

# TEXWRITE

*...a full featured editor/shell*
*designed especially*
*for use with TEX on the PC.*

**Here are some of the things you can do:**

- access TEX, your previewer and device driver without exiting your file
- enter font changes and commands quickly with TEXWRITE'S font and command managers
- edit large files in RAM (supports EMS)
- edit multiple files (you can look at your .LOG file while you make corrections to your .TEX file)
- check for missing braces before running TEX
- write and save text editing functions as macros

TEXWRITE uses pull down menus and function keys for easy access to editor and TEX functions. It's designed to let you be as productive as possible, cutting short the time and keystrokes involved in the edit, TEX, preview and print cycle. As well as providing powerful text editing tools, TEXWRITE provides an integrated, standardized environment for use with TEX.

TEXWRITE Ver. 1.0 will be available the first week of November. It's priced at $149 in Canada, $129 in the U.S., with discounts available for educational and non-profit institutions.

**For more information or to place an order, please contact:**

Docusoft Publishing Technologies Inc.
1150 Homer St., Vancouver, B.C., V6B 2X8, Canada
(604) 687–0354

# DOES YOUR TEX DRIVER RUN OUT OF GAS?

All our DVILASER implementations have a long list of standard features that are hard to find on other TEX drivers. Among them are:

**Font caching.** DVILASER shuffles font information in and out of the printer memory, so there's usually no limit to the number of fonts per page.

**Standard referral to graphic inclusion.** The convention to include graphics and overlays at different magnifications is standard among all our products. If you've used one, you've used them all. Also included in the distribution are sample macros that can be used to reserve space for graphics.

**Condensed .pk font format plus conversion utilities.** Included with DVILASER are the CM fonts in .pk format, plus a wide variety of conversion utilities such as .gf to .pk, .pxl to .pk, and .pl to .tfm.

**Font Scaling.** You can specify a font at *any* size and DVILASER will select the nearest exact representation and magnify or shrink it to the point size you request.

**Many more.** DVILASER supports font substitution, global and local option files, interactive page mode, and more. Call us for a complete list of features not mentioned here.

There are also new standard features on individual versions of DVILASER:

**DVILASER/HP now has HP Soft Font support.** A one-time conversion program creates a .tfm and .pk file from Hewlett-Packard's Soft Fonts. No special downloading is necessary, and they can be scaled to any arbitrary size.

**DVILASER/HP now supports landscape documents.**

**Use any PostScript font with DVILASER/PS.** DVILASER/PS includes an Adobe font metric to TEX font metric conversion utility so you can include future PostScript outline fonts in your documents.

Versions of DVILASER are available for Sun Microsystems, Apollo Domain and VAXStation II workstations; VAX and some IBM mainframes; and IBM PC and compatible microcomputers. IBM PC versions are available through Addison-Wesley (617) 944–6795 and Personal TEX (415) 388–8853.

Why not call or write us and test drive our DVILASER?

**Yes, I would like more information on DVILASER.**

Name:

Title:

Company:

City:

State: Zip:

Phone:

Hardware:

Printer(s):

ARBORTEXT INC. 535 W. William St. Suite 300 Ann Arbor, MI 48103 **(313) 996–3566**

# ...AND YOUR PHOTOTYPESETTER DRIVER?

We don't limit you to printing out your beautiful TEX documents on 300 dpi laser printers; ArborText also offers Autologic and Compugraphic phototypesetter drivers. With them, you can:

**Print "two-up" pages.** You can print any number of DVI pages, each with their own relative horizontal and vertical positioning on a single typesetter film page.

**Produce more documents in less time.** Our optimized drivers never allow redundant typesetter movements even when printing two-up pages.

**Choose printing options interactively.** You can specify a number of options while DVIAPS or DVICG is processing. These include: change default margins and label output pages with your job indentification information.

**Print partial jobs.** You can also rerun a small number of pages if the typesetter film processor destroys a few pages or if typos are caught after the job is run.

**Process pages forward or backward.** You can also process a job forward or backward, and multiple times in a single run.

**Magnify or shrink documents.**

**Display job statistics.** All job statistics including page counts, page dimensions, and paper usage are estimated at the end of the run.

**Complete documentation.** Extensive documentation is included with all our software.

Both software programs are available now for the following typesetters and computer hardware:

**DVIAPS.** DVIAPS supports Autologic phototypesetters. We have versions available that run on DOS, Sun Unix, and VAX/VMS.

**DVICG.** DVICG supports Compugraphic phototypesetters. (8000, 8400, and 9600 require the multi-port option.) A version is available for DOS.

DVIAPS has been used in production environments for over four years and DVICG is currently in use at a number of sites. Why not call or write us for a free sample user manual?

**Yes I would like more information on your phototypesetter drivers.**

Name:

Title:

Company:

City:

State: Zip:

Phone:

Hardware:

Laser Printer(s):

Typesetter:

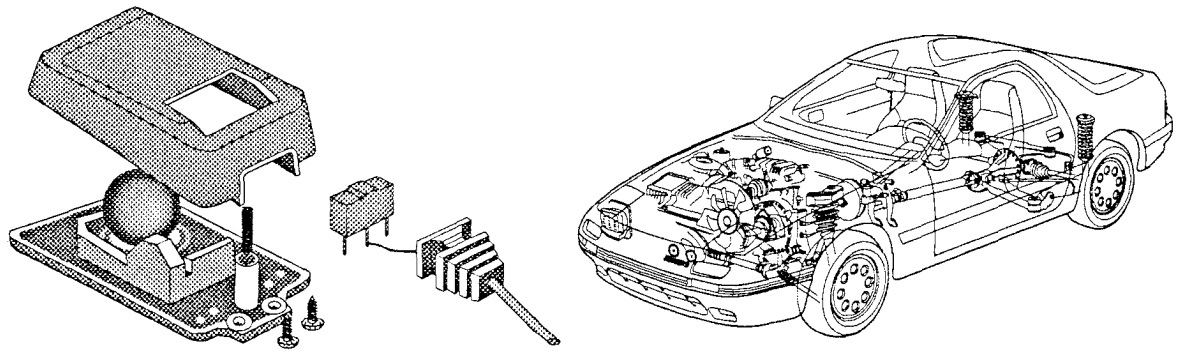
ARBORTEXT INC. 535 W. William St. Suite 300 Ann Arbor, MI 48103 (313) 996–3566

This advertisment was written and formatted on a Sun Workstation using *The Publisher* from ArborText, Inc.
TEX is a trademark of the American Mathematical Society. Sun is a trademark of Sun Microsystems, Incorporated. VAX is a trademark of Digital Equipment Corporation.

# MacTEX™ v2.0

MacTEX v2.0 is faster and better than ever before. It harnesses the power of TEX82 v2.1 under the interactive environment of the Macintosh allowing you to create or edit your TEX files, typeset, and then preview or print them. You can use the mouse in the preview mode to scale and position graphics from the Scrapbook. You can even work with POSTSCRIPT graphics produced with programs such as the Adobe Illustrator, interactively.

MacTEX v2.0 has many advanced features not found in any other TEX system. For example, MacTEX can translate MacWrite or Microsoft WORD files into TEX code automatically, letting you perform *codeless typography*. There is an on-line help facility for PLAIN and LATEX commands. Even the editor has special "TEX" features such as automatic brace matching.



★ TEX82 v2.1 ★ ASCII-based Enhanced Editor with Macro Keys ★ Multi-Page Previewer with four Magnifications ★ Interactive Scrapbook and POSTSCRIPT Graphics Insertion ★ High-Speed POSTSCRIPT Printer Driver ★ Built-in Print Spooler ★ PLAIN and LATEX Formats with Sources ★ INITeX ★ Font Metrics for 98 Adobe Fonts ★ Math ($N = \sum_{k=0}^{n} 4^k = \frac{4}{3}4^n$) and LATEX fonts as downloadable resolution-independant POSTSCRIPT outlines ★ On-line help for PLAIN and LATEX Commands ★ MacWrite and Microsoft WORD-to-TEX Translators ★ DVI-to-POSTSCRIPT Conversion ★ Computer Modern Font Set (Optional) ★ $\mathcal{AMS}$-TEX (Optional) ★ And More... ★

Immediate Availability ★ Price: US$750 (CDN$1000) ★ Site Licences Available

FTL systems Inc.
234 Eglinton Avenue East, Suite 205, Toronto, Ontario CANADA M4P 1K5
Tel: (416) 487-2142, MCI: FTL, Telex: (MCI/WUI) 6502995979 MCI

This Page was Set and Printed Entirely with MacTEX

# TEXT1

Have you ever needed to create a new style sheet for LaTeX, AMS-TeX or Plain TeX, and been frustrated with the work involved in modifying or recoding macros? There is an alternative. *TEXT1*, like AMS-TeX and LaTeX adds to the basic Plain TeX macros such as table of contents, indices, parts, chapters, subtitles, running head and foot text, lists, and auto numbered footnotes and endnotes.

What makes *TEXT1* unique is its *global* formatting macros which give you the ability to easily modify the style sheet (formatting) of each of the local macros, without the need of an experienced TeXnician. For example, if you wanted your chapters to be lettered instead of numbered, or if you wanted to alternate the author's name and chapter title between even and odd pages, you would simply use one global command and change the format. Global format commands let you letter footnotes, or have them print in front of a word instead of following a word. Another global format command allows you to change the list hierarchy from "1., a., i., 1), a), i)" as defined by Turabian to an outline style, "I., A., 1.".

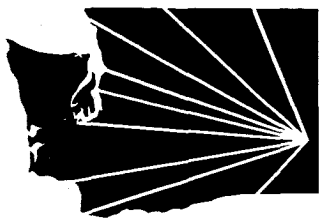
*TEXT1* **has been used by thousands of students, staff and faculty since 1986 to meet their text processing needs.**

*TEXT1* **can be run on any computer that runs TeX.**

*TEXT1* **comes with a complete** *Refrence Manual* **and** *Users Guide*.

*TEXT1* **optionally has available an** *International Phonetic Alphabet* **font in** *GF*, *PK* **or** *PXL* **format. Also available is a** *Compugraphics 8600* **driver.**

## *To order TEXT1, write to*



*TEXT1* Distribution
Computing Service Center
Washington State University
Pullman, WA 99164-1220

## *or call*

*TEXT1* Distribution
(509) 335-0411

**Index of Advertisers**

# Bugs in *Computers & Typesetting*

12 October 1987

This is a list of corrections made to *Computers & Typesetting*, Volumes A–E, since 15 June 87. Corrections made to the softcover version of *The TEXbook* are the same as corrections to Volume A. Corrections to the softcover version of *The METAFONTbook* are the same as corrections to Volume C. Some of these corrections have already been made in reprintings of the books.

A complete list of errata through 15 June 87 can be obtained from the TEX Users Group. For details, write to TUG at P. O. Box 9506, Providence, R.I. 02940-9506, or call (401) 272-9500, ext. 232.

---

Page A326, line 12 (9/20/87)

---

its natural width. The `\hbox` version also invokes `\everypar`.

---

Page A379, line 15 (10/12/87)

---

```
\def\deleterightmost#1{\edef#1{\expandafter\xyzzy#1\xyzzy}}
```

---

Page A454, lines 23–29 (8/13/87)

---

 If a suitable starting letter is found, let it be in font $f$. Hyphenation is abandoned unless the `\hyphenchar` of $f$ is between 0 and 255, and unless a character of that number exists in the font. If this test is passed, TEX continues to scan forward until coming to something that's not one of the following three "admissible items": (1) a character in font $f$ whose `\lccode` is nonzero; (2) a ligature formed entirely from characters of type (1); (3) an implicit kern. The first inadmissible item terminates this part of the process; the trial word consists of all the letters found in admissible items. Notice that all of these letters are in font $f$.

---

Page A462, left column, line 7 (10/9/87)

---

152, 178, 360.

---

Page A473, entry for 'page builder' (8/13/87)

---

when exercised, 122, 280–283, 286–287.

---

Page A481, right column (7/3/87)

---

`\z@`, 347, 348.
`\z@skip`, 347, 348.

---

Page B2, line 32 (9/20/87)

---

**define** *banner* ≡ `'This␣is␣TeX,␣Version␣2.5'` { printed when TEX starts }

---

Page B52, line 5 (8/13/87)

---

cannot be done, i.e., if *hi_mem_min* = *lo_mem_max* + 1, we have to quit.

Page B154, lines 25, 29, 34 respectively (9/20/87)

```
    cvl_backup, radix_backup, co_backup: small_number;  { to save cur_val_level, etc. }
  co_backup ← cur_order; backup_backup ← link(backup_head);
  cur_order ← co_backup; link(backup_head) ← backup_backup;
```

Page B155, new entry for mini-index (9/20/87)

*cur_order*: *glue_ord*, §447.

Page B245, new entry for mini-index (8/7/87)

*cur_s*: *integer*, §616.

Page B254, line 29 (8/7/87)

*cur_s*: *integer*; { current depth of output box nesting, initially −1 }

Page B254, line 31 (8/7/87)

[Remove the statement '*cur_s* ← −1;' and put it on page B244 at the end of line 31.]

Page B266, line 8 (8/7/87)

```
  dvi_out(eop); incr(total_pages); cur_s ← −1;
```

Page B266, new code between lines 31 and 32 (8/7/87)

```
  while cur_s > −1 do
    begin if cur_s > 0 then dvi_out(pop)
    else begin dvi_out(eop); incr(total_pages)
      end;
    decr(cur_s);
    end;
```

Page B338, second-last line (8/19/87)

```
  q ← link(head); s ← head;
```

Page B339, line 4 (8/19/87)

```
    s ← q; q ← link(q);
```

Page B339, new code to insert after line 10 (8/19/87)

```
  if o ≠ 0 then
    begin r ← link(q); link(q) ← null; q ← hpack(q, natural);
    shift_amount(q) ← o; link(q) ← r; link(s) ← q;
    end;
```

[These new lines also imply changes to the index that aren't shown in this errata list.]

Page B547, right column (9/20/87)

*co_backup*: 366.

Page B548, right column (9/20/87)

*cur_order*: 366, 447, 448, 454, 462.

Page B548, right column (8/7/87)

*cur_s*: 593, 616, 619, 629, 640, 642.

Page B559, right column (8/13/87)

[Delete the entry for *low_mem_max*.]

Page B565, left column (8/7/87)

*pop*: 584–585, 586, 590, 601, 608, 642.

Page C26, bottom line (7/18/87)

What angle corresponds to the direction North-Northwest?

Page C107, line 13 (10/7/87)

**pickup penrazor** xscaled *heavyline* rotated (angle($z_{32} - z_{31}$) + 90);

Page C331, just below the illustration (7/18/87)

Such a pattern is, of course, rather unlikely to occur in a `gf` file, but `GFtoDVI` would

Page E32, second-last line (9/20/87)

after which comes '*math_axis#*; **generate** `mathsy`' (which we won't bother to

Page E353, lines 38–39 (8/12/87)

**else: fill** *diag_end*($6r, 5r, 1, 1, 5l, 6l$) -- .9[$z_{5l}, z_{6l}$]
..{$z_5 - z_6$}.1[$z_{5r}, z_{6r}$] -- cycle; % middle stem

Page E387, line 13 (8/12/87)

**pickup** *tiny.nib*; *bulb*(3, 4, 5); % bulb

Page E413, lines 37–38 (8/12/87)

**else: fill** *diag_end*($6r, 5r, 1, 1, 5l, 6l$) -- .9[$z_{5l}, z_{6l}$]
..{$z_5 - z_6$}.1[$z_{5r}, z_{6r}$] -- cycle; % middle stem

Page E459, line 24 (8/7/87)

[Delete the '=' sign between '*lft*' and '$x_5$'.]

Page E485, line 4 (8/7/87)

[Delete the '=' sign between '*lft*' and '$x_5$'.]

Page E550, new line after line 23 (8/15/87)

**forsuffixes** \$ = *notch_cut*, *cap_notch_cut*: **if** \$ < 3: \$ := 3; **fi endfor**

[To make room for this, combine lines 38 and 39 into a single line.]

## Changes to the Programs and Fonts

errdate

### TEX

Changes subsequent to errata publication, 15 June 1987:

```
331. jump_out must fix unfinished output (Found by Klaus Gunterman, 3 Aug 87)
@x module 593
doing_leaders:=false; dead_cycles:=0;
@y
doing_leaders:=false; dead_cycles:=0; cur_s:=-1;
@z
@x module 617
cur_s:=-1; ensure_dvi_open;
@y
ensure_dvi_open;
@z
@x module 640
dvi_out(eop); incr(total_pages);
@y
dvi_out(eop); incr(total_pages); cur_s:=-1;
@z
@x module 642
if total_pages=0 then print_nl("No pages of output.")
@y
while cur_s>-1 do
begin if cur_s>0 then dvi_out(pop)
else begin dvi_out(eop); incr(total_pages);
end;
decr(cur_s);
end;
if total_pages=0 then print_nl("No pages of output.")
@z

332. \hangindent=1pt$$\halign{...\cr\noalign{\hrule}}$$ problem (19 Aug 87)
@x module 805
q:=link(head);
@y
q:=link(head); s:=head;
@z
@x module 805, continued
q:=link(q);
@y
s:=q; q:=link(q);
@z
@x module 806
if is_running(depth(q)) then depth(q):=depth(p);
@y
if is_running(depth(q)) then depth(q):=depth(p);
if o<>0 then
begin r:=link(q); link(q):=null; q:=hpack(q,natural);
shift_amount(q):=o; link(q):=r; link(s):=q;
end;
@z
```

```
333. \hskip 0pt plus 1fil\ifdim problem (found by Alan Guth, 20 Aug 87)
@x module 366
@!cvl_backup,@!radix_backup:small_number; {to save |cur_val_level| and |radix|}
@y
@!cvl_backup,@!radix_backup,@!co_backup:small_number;
{to save |cur_val_level|, etc.}
@z
@x
backup_backup:=link(backup_head);
@y
co_backup:=cur_order; backup_backup:=link(backup_head);
@z
@x
link(backup_head):=backup_backup;
@y
cur_order:=co_backup; link(backup_head):=backup_backup;
@z

334. (I sincerely hope that there won't be any more)
```

METAFONT

Changes subsequent to publication of errata list, 15 Jun 1987:

No later changes.

### Computer Modern fonts

Changes subsequent to publication of the full errata list, 15 June 1987:

```
@x in SYM, the plus-or-minus character
x1=x2=.5w; lft x3=lft=x5=hround u-eps; x4=x6=w-x3;
@y
x1=x2=.5w; lft x3=lft x5=hround u-eps; x4=x6=w-x3;
@z actually the code worked but it was "infelicitous"

@x in SYMBOL, the minus-or-plus character
x1=x2=.5w; lft x3=lft=x5=hround u-eps; x4=x6=w-x3;
@y
x1=x2=.5w; lft x3=lft x5=hround u-eps; x4=x6=w-x3;
@z actually the code worked but it was "infelicitous"

@x in ROMANU, letter J [fixes a bug if dish=0 and crisp<tiny and serifs]
 bulb(3,4,5); % bulb
@y
 pickup tiny.nib; bulb(3,4,5); % bulb
@z

@x in ROMANL, letter w [makes notch_cut more useful]
 else: fill diag_end(6r,5r,1,1,5l,6l)--.5[z5l,z6l]
   --.5[z5r,z6r]--cycle;% middle stem
@y
 else: fill diag_end(6r,5r,1,1,5l,6l)--.9[z5l,z6l]
   ..{z5-z6}.1[z5r,z6r]--cycle; % middle stem
@z the same change applies also to letter W in ROMANU

@x in CMBASE, makes lowres types (especially TT) look better
 define_blacker_pixels(notch_cut,cap_notch_cut);
@y
 define_blacker_pixels(notch_cut,cap_notch_cut);
 forsuffixes $=notch_cut,cap_notch_cut: if $<3: $:=3; fi endfor
@z

(I sincerely hope there won't be any more!)
```

# TEX Users Group Membership List — Supplement

Compiled 15 October 1987

This list includes the names of all persons who have become members of TUG or whose addresses have changed between the time the list was last compiled (15 June 1987) and 15 October 1987. If a name does not appear in this supplement, check the list distributed with the first issue of the year and the supplement published in issue no. 2. All institutional members are listed. Total membership: 129 institutional members and 2,758 individuals affiliated with more than 1,000 colleges and universities, commercial publishers, government agencies, and other organizations throughout the world having need for an advanced composition system. One more complete membership list will be issued before the end of 1987.

The following information is included for each listing of an individual member, where it has been provided:

Name and mailing address

Telephone number

Network address

Title and organizational affiliation, when that is not obvious from the mailing address

Computer and typesetting equipment available to the member, or type of equipment on which his organization wishes to (or has) installed TEX

Uses to which TEX may be put, or a general indication of why the member is interested in TEX

CONTENTS



Recipients of this list are encouraged to use it to identify others with similar interests, and, as TUG members, to keep their own listings up-to-date in order for the list to remain as useful as possible. New or changed information may be submitted on the membership renewal form bound into the back of a recent issue of TUGboat. Comments on ways in which the content and presentation of the membership list can be improved are welcome.

This list is intended for the private use of TUG members. Additional copies are available to members on request. Mailing lists of current TUG membership are available for purchase. For more information, contact Ray Goucher, TUG Business Manager, at (401) 272-9500, ext. 232.


Distributed with TUGboat Volume 8 (1987), No. 3. Published by

**TEX Users Group**
P. O. Box 9506
Providence, R.I. 02940-9506, U.S.A.

# Addresses of TUG Steering Committee Members and Members of Other TUG Committees

## Site Coordinators

### • CDC Cyber

**FOX, Jim**
Academic Computing Center HG-45
University of Washington
3737 Brooklyn Ave NE
Seattle, WA 98105
206-543-4320
Bitnet: C4328@uwarita
Bitnet: fox7632@uwacdc

### ••DG MV

**CHILDS, S. Bart**
Dept of Computer Science
Texas A & M University
College Station, TX 77843-3112
409-845-5470
Bitnet: Bart@TAMLSR
President; Finance Committee

### • IBM MVS

**PLATT, Craig R.**
Dept of Math & Astronomy
Machray Hall
Univ of Manitoba
Winnipeg R3T 2N2, Manitoba Can
204-474-9832
CSnet: platt@uofm.cc.cdn
Bitnet: platt@uofmcc

### • IBM VM/CMS

**GUENTHER, Dean**
Computer Service Center
Washington State University
Computer Science Building, Room 2144
Pullman WA 99164-1220
509-335-0411
Bitnet: Guenther@WSUVM1

### ••Prime 50 Series

**CRAWFORD, John M.**
Computing Services Center
College of Business
Ohio State University
1775 College Road
Columbus, OH 43210
614-292-1741
Arpanet: Crawford-J@Ohio-State
Bitnet: TS0135@OHSTVMA
International Coordinator

### •• "small" systems

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

### ••UNIX

**MacKAY, Pierre A.**
Northwest Computer Support Group
University of Washington
Mail Stop DW-10
Seattle, WA 98195
206-543-6259
Arpanet:
MacKay@June.CS.Washington.Edu

### ••VAX (VMS)

**SMITH, Barry**
Kelierman & Smith
534 SW Third Avenue
Portland, OR 97204
503-222-4234; TLX 9102404397
Usenet: tektronix!reed!barry

## Officers

**CHILDS, S. Bart**
President; Finance Committee;
DG MV Site Coordinator

**THEDFORD, Rilla**
Intergraph Corporation
MS HQ1006
One Madison Industrial Park
Huntsville, AL 35807
205-772-2440
Vice President; Finance Committee

**HOENIG, Alan**
17 Bay Avenue
Huntington, NY 11743
516-385-0736
Secretary; Finance Committee

**NESS, David**
TV Guide
Radnor, PA 19088
215-293-8860
Treasurer; Finance Committee

## Other Steering Committee Members

**BECK, Lawrence A.**
Grumman Data Systems
R & D, MS D12-237
Woodbury, NY 11797
516-682-8478
X3V1.8 Coordinator

**BEETON, Barbara**
American Mathematical Society
P. O. Box 6248
Providence, RI 02940
401-272-9500 x299
Arpanet: BNB@Seed.AMS.com
BNB@XX.LCS.MIT.edu
Editor, TUGboat

**DYER, Allen R.**
2922 Wyman Parkway
Baltimore, MD 21211
301-243-0008 or 243-7283

**FUCHS, David**
1775 Newell
Palo Alto, CA 94303
415-323-9436

**FURUTA, Richard**
Department of Computer Science
University of Maryland
College Park, MD 20742
301-454-1461
Arpanet: furuta@mimsy.umd.edu
Laser-Lovers moderator;
Finance Committee

**GOUCHER, Raymond**
TEX Users Group
P. O. Box 9506
Providence, RI 02940-9506
401-272-9500 x232
Arpanet: reg@Seed.AMS.com
Executive Director

**HENDERSON, Doug**
Division of Library Automation
Univ of California, Berkeley
186 University Hall
Berkeley, CA 94720
415-642-9485
Bitnet: dlatex@ucbcmsa
Metafont Coordinator

**ION, Patrick D.**
Mathematical Reviews
416 Fourth Street
P. O. Box 8604
Ann Arbor, MI 48107
313-996-5273
Arpanet: ion@Seed.AMS.com

**KNUTH, Donald E.**
Department of Computer Science
Stanford University
Stanford, CA 94305
Arpanet: DEK@Sail.Stanford.edu

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

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

**WHIDDEN, Samuel B.**
American Mathematical Society
P. O. Box 6248
Providence, RI 02940
401-272-9500
Arpanet: sbw@seed.ams.com
Finance Committee

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

## Members of Other Committees

### ••Annual Meeting Program Committee

**GUENTHER, Dean R.**
Computer Service Center
Washington State University
Computer Science Bldg
Pullman, WA 99164
509-335-0411
Bitnet: GUENTHER@WSUVM1

### ••Committee on Corporate Relationships

**FERGUSON, Michael J.**
INRS - Télécommunications
Univ du Québec
3 Place du Commerce
Verdun (H3E 1H6), Québec Canada
514-765-7834
CSnet: mike%tel.inrs.cdn@ubc.csnet

**SMITH, Barry**

**WHIDDEN, Samuel B.**

### •• Nominating Committee

**FERGUSON, Michael J.**

**ION, Patrick D.**

**JACKSON, Calvin W., Jr.**
1749 Micheltorena St
Los Angeles, CA 90026
818-356-6245 or 213-660-3257

### ••1988 Scholarship Committee

**HSIA, Doris T.**
2630 Sierra Vista Ct
San Jose, CA 95116
415-723-8117

**McPARTLAND-CONN, Marie**
GTE Laboratories
40 Sylvan Rd
Waltham, MA 02254
617-466-4220

### ••TUGboat Editorial Committee

**BEETON, Barbara**
Editor

**DAMRAU, Jackie**
Dept of Math & Statistics
Univ of New Mexico
Albuquerque, NM 87131
505-277-4623
Bitnet: damrau@unmb
UUCP: damrau@unmvax
Associate Editor, LATEX

**EPPSTEIN, Maureen**
Administrative Publications
Stanford University
Encina Hall, Room 200
Stanford, CA 94305
415-725-1717
Arpanet:
as.mve@Forsythe.Stanford.Edu
Associate Editor for Applications

**HOENIG, Alan**
Associate Co-Editor, Typesetting on Personal Computers

**JÜRGENSEN, Helmut**
Dept of Computer Science
Univ of Western Ontario
London N6A 5B7, Ontario, Canada
519-661-3560
Bitnet: A505@UWOCC1
UUCP: helmut@deepthot
Associate Editor for Software

**MANN, Laurie D.**
Stratus Computer
55 Fairbanks Blvd
Marlboro, MA 01752
617-460-2610
uucp: harvard!anvilles!Mann
Associate Editor for Training issues

**PFEFFER, Mitch**
Suite 90
148 Harbor View South
Lawrence, NY 11559
516-239-4110
Associate Co-Editor, Typesetting on Personal Computers

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

Addresses and telephone numbers of individuals serving in more than one capacity are listed only once.

# TUG Institutional Members

**Addison-Wesley Publishing Company**
Reading, Massachusetts

**The Aerospace Corporation**
El Segundo, California

**American Mathematical Society**
Providence, Rhode Island

**ArborText, Inc.**
Ann Arbor, Michigan

**ASCII Corporation**
Tokyo, Japan

**Aston University**
Birmingham, England

**California Institute of Technology**
Pasadena, California

**Calvin College**
Grand Rapids, Michigan

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

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

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

**College of William & Mary**
Department of Computer Science
Williamsburg, Virginia

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

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

**Data General Corporation**
Westboro, Massachusetts

**DECUS, L&T Special Interest Group**
Marlboro, Massachusetts

**Department of National Defence**
Ottawa, Ontario, Canada

**Digital Equipment Corporation**
Nashua, New Hampshire

**dit Company, Ltd.**
Tokyo, Japan

**Edinboro University of Pennsylvania**
Edinboro, Pennsylvania

**Electricité de France**
Clamart, France

**Environmental Research Institute of Michigan**
Ann Arbor, Michigan

**European Southern Observatory**
Garching bei München, Federal Republic of Germany

**Ford Aerospace & Communications Corporation**
Palo Alto, California

**Försvarets Materielverk**
Stockholm, Sweden

**General Motors Research Laboratories**
Warren, Michigan

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

**Grinnell College**
Computer Services
Grinnell, Iowa

**Grumman Corporation**
Bethpage, New York

**GTE Laboratories**
Waltham, Massachusetts

**Hart Information Systems**
Austin, Texas

**Hartford Graduate Center**
Hartford, Connecticut

**Harvard University**
Computer Services
Cambridge, Massachusetts

**Hewlett-Packard Co.**
Boise, Idaho

**Hobart & William Smith Colleges**
Geneva, New York

**Humboldt State University**
Arcata, California

**Hutchinson Community College**
Hutchinson, Kansas

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

**Illinois Institute of Technology**
Chicago, Illinois

**Imagen**
Santa Clara, California

**Institute for Advanced Study**
Princeton, New Jersey

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

**Intergraph Corporation**
Huntsville, Alabama

**Intevep S. A.**
Caracas, Venezuela

**Iowa State University**
Ames, Iowa

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

**Kuwait Institute for Scientific Research**
Safat, Kuwait

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

**Louisiana State University**
Baton Rouge, Louisiana

**Marquette University**
Department of Mathematics, Statistics
and Computer Science
Milwaukee, Wisconsin

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

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

**Max Planck Institute Stuttgart**
Stuttgart, Federal Republic of Germany

**McGill University**
Montreal, Quebec, Canada

**National Institutes of Health**
Bethesda, Maryland

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

**New Jersey Institute of Technology**
Newark, New Jersey

**New York University**
Academic Computing Facility
New York, New York

**Northeastern University**
Academic Computing Services
Boston, Massachusetts

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

**Pennsylvania State University**
Computation Center
University Park, Pennsylvania

**Personal TEX, Incorporated**
Mill Valley, California

**Purdue University**
West Lafayette, Indiana

**QMS, Inc**
Mobile, Alabama

**Queens College**
Flushing, New York

**Research Triangle Institute**
Research Triangle Park, North Carolina

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

**Ruhr Universität Bochum**
Bochum, Federal Republic of Germany

**Rutgers University**
Hill Center
Piscataway, New Jersey

**St. Albans School**
Mount St. Alban, Washington, D.C.

**Sandia National Laboratories**
Albuquerque, New Mexico

**SAS Institute**
Cary, North Carolina

**Schlumberger Well Services**
Houston, Texas

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

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

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

**Software Research Associates**
Tokyo, Japan

**Sony Corporation**
Atsugi, Japan

**Space Telescope Science Institute**
Baltimore, Maryland

**Springer-Verlag**
Heidelberg, Federal Republic of Germany

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

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

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

**State University of New York**
Department of Computer Science
Stony Brook, New York

**Stratus Computer, Inc.**
Marlboro, Massachusetts

**Syracuse University**
Syracuse, New York

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

**Texas A & M University**
Computing Services Center
College Station, Texas

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

**TRW, Inc.**
Redondo Beach, California

**Tufts University**
Medford, Massachusetts

**TV Guide**
Radnor, Pennsylvania

**UNI.C**
Danmarks EDB-Center
Aarhus, Denmark

**University of Alabama**
Tuscaloosa, Alabama

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

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

**University of Calgary**
Calgary, Alberta, Canada

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

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

**University of California, San Diego**
La Jolla, California

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

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

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

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

**University of Delaware**
Newark, Delaware

**University of Glasgow**
Glasgow, Scotland

**University of Groningen**
Groningen, The Netherlands

**University of Illinois at Chicago**
Computer Center
Chicago, Illinois

**University of Kansas**
Academic Computing Services
Lawrence, Kansas

**University of Maryland**
College Park, Maryland

**University of Massachusetts**
Amherst, Massachusetts

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

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

**University of Ottawa**
Ottawa, Ontario, Canada

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

**University of Tennessee at Knoxville**
Department of Electrical Engineering
Knoxville, Tennessee

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

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

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

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

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

**Vanderbilt University**
Nashville, Tennessee

**Vereinigte Aluminium-Werke AG**
Bonn, Federal Republic of Germany

**Villanova University**
Villanova, Pennsylvania

**Washington State University**
Pullman, Washington

**Widener University**
Computing Services
Chester, Pennsylvania

**John Wiley & Sons, Incorporated**
New York, New York

**Worcester Polytechnic Institute**
Worcester, Massachusetts

**Yale University**
Department of Computer Science
New Haven, Connecticut

(129 institutional members as of 25 Oct 1987)