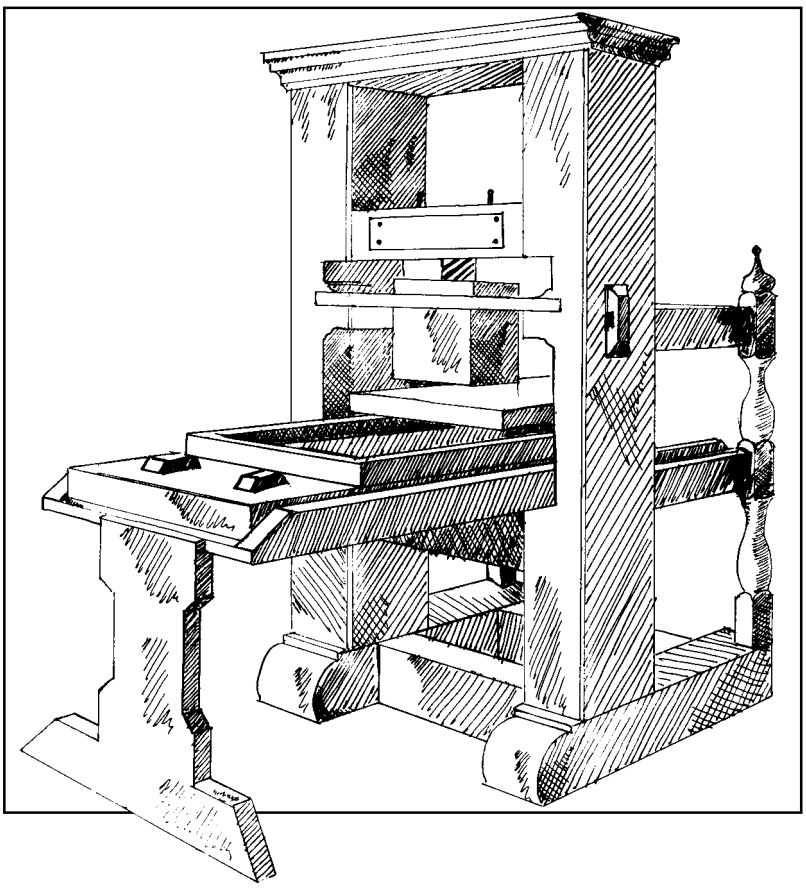
TUGboat

Volume 18, Number 2 / June 1997

TUGBOAT

The Communications of the TEX Users Group



Volume 18, Number 2, June 1997

TEX Users Group

Memberships and Subscriptions

TUGboat (ISSN 0896-3207) is published quarterly by the TEX Users Group, P. O. Box 1239, Three Rivers, CA 93271-1239, U.S.A.

1997 dues for individual members are as follows:

- Ordinary members: $60.
- Students: $40.

Membership in the TEX Users Group is for the calendar year, and includes all issues of *TUGboat* for the year in which membership begins or is renewed. Individual membership is open only to named individuals, and carries with it such rights and responsibilities as voting in the annual election. A membership form is provided on page 142.

TUGboat subscriptions are available to organizations and others wishing to receive *TUGboat* in a name other than that of an individual. Subscription rates: $70 a year, including air mail delivery.

Periodical-class postage paid at San Francisco, CA, and additional mailing offices. Postmaster: Send address changes to *TUGboat*, TEX Users Group, P. O. Box 1239, Three Rivers, CA 93271-1239, U.S.A..

Institutional Membership

Institutional Membership is a means of showing continuing interest in and support for both TEX and the TEX Users Group. For further information, contact the TUG office.

Printed in U.S.A.

Board of Directors

Addresses
All correspondence, payments, parcels, etc.
TEX Users Group
P. O. Box 1239
Three Rivers,
CA 93271-1239 USA

To obtain an address suitable for courier or delivery services, call or email the TUG office.

Telephone
+1 209 561-0112

Fax
+1 209 561-4584

Electronic Mail
(Internet)
General correspondence:
`TUG@mail.tug.org`
Submissions to *TUGboat*:
`TUGboat@mail.tug.org`

World Wide Web
`http://www.tug.org/`
`http://www.tug.org/TUGboat/`

. . . in May 1977 I didn't know what to do about fonts;
June 1977 is when I started to have the idea of METAFONT.

Donald E. Knuth
Prague, March 1996
Questions and answers with $\mathcal{CS}$TUG

TUGBOAT

COMMUNICATIONS OF THE TEX USERS GROUP

EDITOR BARBARA BEETON

VOLUME 18, NUMBER 2 • JUNE 1997
THREE RIVERS • CALIFORNIA • U.S.A.

TUGboat

During 1997, the communications of the TEX Users Group will be published in four issues. Issue 3 will contain the Proceedings of the 1997 TUG Annual Meeting.

TUGboat is distributed as a benefit of membership to all members.

Submissions to *TUGboat* are reviewed by volunteers and checked by the Editor before publication. However, the authors are still assumed to be the experts. Questions regarding content or accuracy should therefore be directed to the authors, with an information copy to the Editor.

Submitting Items for Publication

The next regular issue will be Vol. 18, No. 4. The deadline for technical items is October 20; reports and similar items are due by November 10. (Items for the Calendar of the Proceedings issue should be received by August 25.) Mailing is scheduled for December. Deadlines for other future issues are listed in the Calendar, page 138.

Manuscripts should be submitted to a member of the *TUGboat* Editorial Board. 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 (see address on p. 67).

Contributions in electronic form are encouraged, via electronic mail, on diskette, or made available for the Editor to retrieve by anonymous FTP; contributions in the form of camera copy are also accepted. The *TUGboat* "style files", for use with either `plain` TEX or LATEX, are available "on all good archives". For authors who have no network FTP access, they will be sent on request; please specify which is preferred. Write or call the TUG office, or send e-mail to `TUGboat@mail.tug.org`.

This is also the preferred address for submitting contributions via electronic mail.

Reviewers

Additional reviewers are needed, to assist in checking new articles for completeness, accuracy, and presentation. Volunteers are invited to submit their names and interests for consideration; write to `TUGboat@mail.tug.org` or to the Editor, Barbara Beeton (see address on p. 67).

TUGboat Advertising and Mailing Lists

For information about advertising rates, publication schedules or the purchase of TUG mailing lists, write or call the TUG office.

TUGboat Editorial Board

See page 67 for addresses.

Other TUG Publications

TUG publishes the series *TEXniques*, in which have appeared reference materials and user manuals for macro packages and TEX-related software, as well as the Proceedings of the 1987 and 1988 Annual Meetings. Other publications on TEXnical subjects also appear from time to time.

TUG is interested in considering additional manuscripts for publication. These might include manuals, instructional materials, documentation, or works on any other topic that might be useful to the TEX community in general. Provision can be made for including macro packages or software in computer-readable form. If you have any such items or know of any that you would like considered for publication, send the information to the attention of the Publications Committee in care of the TUG office.

Trademarks

Many trademarked names appear in the pages of *TUGboat*. If there is any question about whether a name is or is not a trademark, prudence dictates that it should be treated as if it is. The following list of trademarks which appear in this issue may not be complete.

MS/DOS is a trademark of MicroSoft Corporation
METAFONT is a trademark of Addison-Wesley Inc.
PC TEX is a registered trademark of Personal TEX, Inc.
PostScript is a trademark of Adobe Systems, Inc.
TEX and $\mathcal{AMS}$-TEX are trademarks of the American Mathematical Society.
Textures is a trademark of Blue Sky Research.
UNIX is a registered trademark of X/Open Co. Ltd.

Letters

Letter from a member

Phil Kopriva

Dear Michel, 17 April 1997

Please let me take this opportunity to thank you for your time and effort spent on the behalf of TUG, as a board member, Vice-President and President, for the past 3 years. We all appreciate the effect your presence provided. Your guidance will be missed. Thank you, again.

After reading your "The State of TUG" article in the latest *TUGboat* (Vol. 17, No. 4 December 1996) I feel compelled to make a few comments. In general, I must agree with your observations, but, as a TEX novice and 6–7 year member of TUG, I would like to respond to some of your speculations regarding the diminishing membership.

To let you know something about myself, I was introduced to TEX by Dan Levin, here in Northern California. At that time I was interested in typesetting mathematical statements, for educational purposes. I am now retired from teaching and research (Physiology/Biophysics), so, at this point, my use of TEX is strictly personal. I'm currently working on a book (using LATEX) dealing with the mathematics of investments, for those people who have forgotten, or never really learned, their mathematics. I have essentially no contact with other TEX users, so my level of expertise is questionable, if not wholly deficient. Having described myself somewhat, the following comments may be more understandable.

"Computer-type" people seem to speak a language quite different from us "regular" type! My point, since I've been receiving *TUGboat*, the over usage of ACRONYMS has greatly lessened my enjoyment and the usefulness of the various articles. Acronyms should *always* be defined when first used in an article. In fact, I recall a macro which does exactly that. An author/editor should never assume that we readers know all the acronyms used. Maybe there could be half of a page, in each *TUGboat* edition, dedicated to commonly used acronyms and their definitions. Philip Taylor, author of the **Survey** article (p. 367), in the abovementioned issue of *TUGboat*, dealt with this problem very well.

Several years ago the decision was made to discontinue, or at least, combine the smaller *TEX and TUG NEWS* (TTN), with the *TUGboat*. I understand the economics of this decision, however, if there were many members of my calibre, I would not be surprised if that decision was partially responsible for driving some of them away. The statement, at the time, was that columns such as "Typographer's Inn", "Hey — it works!" and "LATEX News" would continue, maybe not with the same titles, authors or format, but, something from which we novices could learn. To date, this has occurred only minimally!! *TUGboat* is apparently written for the expert TEX and LATEX user. I find very little any more which I can understand, let alone use. This is certainly a testimonial to my computerese, but, is also, possibly partially responsible for the decline in some of the membership. The board members and authors must remember that most of the membership probably do not work in University computer centers, nor are we computer and TEX experts. We are novices, plugging away, at home, on our desktop PC's.

You address the issue of the impact of electronic communications. My quest is for macros which will help me with my enjoyment and application of TEX.

Along with *acronyms*, *TUGboat* could list good WEB sites and TEX User Groups for the readers. A useful publication should not make us "dig" out this information!

My point is this, unless TUG directors begin addressing the great unwashed TEX/LATEX user, membership will probably continue to decrease. I don't think membership cost ($) is terribly important to most of us, but getting something useful from you TEX-gurus is. There is little to no help, from either the written or electronic press.

If I sound frustrated, I am. I love TEX and LATEX and will continue to support TUG, but give me something in return. I generally contribute $ well above and beyond the regular dues. I have all the texts and books, including yours, but have neither the ability, desire or time to "reinvent the wheel". What I do want is to be able to pick and choose from the minds of others. Sounds selfish and lazy? Maybe, but that's how I feel at this point. Yes, I do appreciate the work done by the early pioneers, but that ground has already been plowed.

I've taken enough of your time. Please consider some of my suggestions and pass them on to whomever might make them happen. Although I spend much of the day in a wheelchair, maybe I'll see you at the conference this summer.

Thank you,

◇ Phil Kopriva
San Francisco
phil.kopriva@ggcs.org

General Delivery

From the President

Michel Goossens

It was just over two years ago that I was elected President of the TEX Users Group, taking over from Christina Thiele at TUG'95 in St. Petersburg. During both our terms the basic shift away from mainframes to personal computers, away from a multi-million dollar infrastructure limited to university departments, and large laboratories or companies to the desks of secretaries and students was completed. Nowadays, our sons and daughters all have a personal computer in their room, where it assists them with their homework, allows them to listen to and view the far-away concert of their favorite pop group on the Internet, lets them delve into the depths of knowledge thanks to thousands of thematic educational web-sites and CDROM's. World events invade our homes via coaxial cable or parabolic dish, digital technology projects hundreds of television channels of perfect quality on our television screens, cybershops are popping up on every street-corner, taking our credit card number and delivering a pizza and diet Coke in less than no time to our home.

We really have entered the information age, and the only drawback is that we must be careful not to drown in an ocean of colorful images and surround-sound music. One of our main tasks is to find ways to put some order in this relatively chaotic flood of megabits, to present our brains with patterns that it can recognize, handle and store reliably for later retrieval. Knowledge is information with a recognizable structure, something we can understand, classify and apply in the future. But knowledge is useless unless it is shared, thus stressing the importance of efficient communication. A yard worth of encyclopaedia on the shelves in our study is orders of magnitude less useful than the content of that same encyclopaedia stored on the Internet, where it can potentially be consulted by anybody anytime anywhere in the world. Communicating structured information is the key to optimal transmission of ideas, the basis of human society and culture. Access to Internet allows otherwise isolated people — physically handicapped, visually impaired, those living in remote areas — to become a genuine part of the cyberworld, thus breaking their isolation, offering them emancipation and allowing them to communicate with people everywhere, in one word giving them a passport to the world at large.

We, the TEX community, are a small but important part of this global information infrastructure. Thanks to the genius of Donald Knuth we are able to communicate mathematical ideas precisely, concisely, beautifully, yet typographically correct. Well before the word Internet became fashionable Prof. Knuth realized the need to develop and define a language to capture the precise meaning of the complex essentially two-dimensional mathematical notation into a linear sequence of tokens that can be understood relatively easily by human writers and interpreted by computer code. Rarely in the course of human history has a single invention yielded so high a technological payoff in such a short time. The TEX language, and its structured variant LATEX, allow authors to communicate their mathematical reasoning and results precisely and unambiguously to collaborators, students, and publishers in the remotest corners of the world with the speed of the bit-stream carrying the electronic message on the Internet.

Millions of people worldwide have used TEX in the past, and will use it in the future, be it explicitly, by typing backslashes preceding TEX keywords, or indirectly, by using systems which rely on TEX as their typesetting engine. It is TUG's role to ensure that the work of Knuth will live well into the next century, so that future generations of students and researchers, poets and religious and other scholars can benefit from it.

However, we should not stand still and merely consolidate past developments, but we should actively stimulate further work to extend TEX's superb mathematical typesetting capabilities, combine them with hypertext and multi-byte multi-lingual functions. With his characteristic enthusiasm Knuth endorses such investigations, and recent initiatives such as ε-TEX, Ω, and `pdftex` have his full support.

As set out in Article One of TUG's bylaws, we should not only concentrate on TEX *per se*, but we should look at typesetting technical text and font design, and exchange of (technical) information in general. Therefore we should be aware of the activity of the World Wide Web Consortium (W3C) in the area of XML (Extensible Markup Language). XML differs from HTML (Hypertext Markup Language) in several ways, the more important being that the author can extend the basic SGML (Standard Generalized Markup Language, ISO 8879) by defining new tags and attribute names. A second important development is the introduction of style sheets to guide the hypertext viewing programs in rendering the information. Two aspects to note are the cascading style sheets (CSS), which is probably

sufficient for handling current level HTML, and the more powerful Document Style Semantics and Specification Language (DSSSL, ISO/IEC 10179), which is a perfect complement for XML. In particular, an implementation based on James Clark's DSSSL engine Jade with a back-end for LaTeX (`jadetex`) has become available. Also, on May 15th a working draft of MathML (Mathematical Markup Language) was released. MathML is an XML application for describing structure and content of mathematical expressions, so that mathematics can be reliably handled on the Web. This collaborative effort is of great significance for the scientific community. The fact that all major players in the field (symbolic algebra developers, math societies, browser implementors, scientific publishers) could come to a common proposal shows the importance of this issue. A final standard (including fonts, character sets, operator names) is expected by mid-1998, but first implementations in renderers like the Geometry Center's *WebEQ* and IBM's `techexplorer` are expected soon.

At the same time we should also not forget the contributions by many hundreds of developers of TeX and LaTeX classes, packages and extensions, or the thousands of maintainers of local TeX systems in numerous sites dispersed over the earth's surface. TeX, because it is free, and relatively straightforward to use and install, has become an integral part of the working culture of mathematicians, physicists, and other scientists the world over. TeX can run on the smallest personal computers, but also many full-blown commercial and free applications take advantage of the richness of the TeX composition model: Mathematica, Scientific Office, Y&Y, TrueTeX, Lyx, GNU's documentation system are but a few recent examples of the use of TeX directly or indirectly.

Nowadays the basic TeX setup is available as a plug-and-play runnable system for most operating systems on CDROMs or from one of the CTAN sites on the Internet. In fact, this issue of *TUGboat* contains Version 2 of the *TeX Live* CDROM, which is based on the latest Web2C version of TeX and supports many Unix platforms, Windows 95 and NT, Amiga. It also has a distribution for MS-DOS and Macintosh. The *TeX Live* CDROM is a joint initiative of TUG, GUTenberg, and UKTUG, with full support of other TeX User Groups. I hope that together with our continuous efforts to develop our own Internet Web and ftp sites (`www.tug.org` and `ftp.tug.org`) this shows that TUG takes the Internet and electronic media seriously as we consider them as convenient vehicles to optimize communication between information and software providers and their users.

Of course, TUG and the TeX community at large rely on the continuous and generous support and collaboration of a large network of volunteers. After having mentioned ongoing initiatives in the previous sections, in this my last editorial as TeX President, I would not like to miss the occasion to say a word about my colleagues on the TUG Board. In the last issue of *TUGboat* I welcomed the incoming members to the Board. For the next four years (1997–2001) it will be up to them to steer TUG successfully into the next century and complete the transition to an Internet-based strategy, which was started two years ago. Therefore, this time I want to pay a special tribute to the outgoing members of the TUG Board, those who for many years have given the best of themselves in the interest of the international TeX Community. Thank you Mimi Burbank, Robin Fairbairns, George Greenwade, Yannis Haralambous, John Radel, and Sebastian Rahtz for the many contributions in your respective areas of competence. In the name of the whole of TUG I wish you well in your careers and all the best for you and your families. And then there are four present Board members who continue. Barbara Beeton — who has been at the very heart of TUG, *TUGboat* (and TeX!) from the very beginning, Karl Berry, who can pride himself on having adapted TeX to mainstream computing, so that it can be easily installed on almost any computer platform in the world with minimal effort, Judy Johnson, whose background in publishing comes in handy, and Jiří Zlatuška, who plays an important role in the NTS, `pdftex`, and ε-TeX developments. Thanks to their years-long experience I rest assured that they will be an invaluable asset for the new TUG Board and provide the continuity needed to stimulate, guide and finalize the transition of TUG's role and the redefinition and optimization of the function of the TUG Office.

And last but not least, I want to address myself to you personally, dear reader, and TUG member. Thank you for your continued support. I know that these last few years TUG has been confronted with a dwindling membership. The causes have been debated at length in this column and elsewhere. Notwithstanding this difficult situation, with *TUGboat* arriving late, the responsiveness of the TUG Office not so quick and effective as it could be, you have remained loyal and it is thanks to your continued faith in us that TUG is still alive, not just as an aim in itself, but to promote and guard Knuth's contribution to mankind. That is what we

should do, and will continue to do as long as we can. I am sure that if we can count on your help, TUG, and especially TEX, still have a long and healthy future.

Response to Mr. Kopriva

Dear Phil,

Let me start by thanking you for your continued support of TUG. As I explained in the last paragraph of my editorial above, it is thanks to members like you who believe that TUG still has a future, that the Board and *TUGboat* production team can continue do their best to serve the TEX community at large.

I completely agree that acronyms should *always* be defined when they first occur. But, for authors, it is not always evident that they in fact use an acronym at all. Very often one uses TV (for television) or, to take an example from biology, DNA (for *deoxyribonucleic acid*) without spelling out what it means, because it is assumed — wrongly, perhaps — that readers of the newspaper, journal, or book "know" what it means. We shall, however, do our best to publish a list of often-occurring acronyms in each issue of *TUGboat* and to define them at least once in every article (as I tried to do for the important ones in the editorial).

You also mention the decision we took some time ago to discontinue *TTN* (*TEX and TUG News*), and fold its content into *TUGboat*. You point out, correctly, that some of the columns, like "Typographer's Inn", "Hey — it works!", and others have not yet reappeared. Barbara Beeton, the *TUGboat* Editor, has spent quite a bit of time contacting the (previous, *TTN*) columnists. Although they informed her that they have some material available, they are all busy with other pursuits at the moment, and unable to deliver anything for publication. This is a sad state of affairs, but we have to face the fact that nobody has been found to take on the job of running these columns since early 1995. In *TUGboat* 14#4 (the December 1993 issue, thus well before *TTN* was discontinued), Barbara came up with with a "*TUGboat* wish list" (see page 373). She asked explicitly for authors, or other volunteers. That was over three years ago, and since then only two "semi" volunteers have come forward, but only to review items, not ready to write or take action by "shaking the bushes" and going after other potential volunteers. At the same time some of the other functions of *TTN*, such as news about TEX-related activities and meetings, reviews of articles appearing in the magazines of other TEX Groups, reviews of books, have been published regularly in *TUGboat*.

One area where a volunteer could be extremely useful is to read the newsgroup `comp.text.tex` regularly, identify "hot" topics, and summarize the discussions in those areas. All we can do today is to spot contributions which we consider particularly well done, and then try to invite the author(s) in question to write it up for *TUGboat*. Sadly enough, several such queries have remained without replies at the moment, although we are still hoping for positive responses. The problem is that most members of the *TUGboat* team have little time to track such queries regularly, write to the authors (several times, if necessary), help them to collect the material, put it in a digestable form, etc. It would be useful if you yourself, or another of our readers, could tell us about possible candidates who have the time and energy to take on this very important task.

And this brings me, once more, to emphasize the fact that none of the members of the *TUGboat* team (nor of the Board) get paid to do *TUGboat* or anything related to TUG. In most cases, our respective employers simply "tolerate" that we work for TUG and *TUGboat* outside normal working hours and on weekends (and, as you can imagine, our families do not always share our enthusiasm for TEX!).

In the area of pointers to good Web sites and TEX User Groups, please take a look at TUG's own Web site at the URL `http://www.tug.org/`. We have recently put quite some effort in collecting there a lot of TEX-related information, and if you are still missing something, please let us know so that we can add it. We should also mention that in 1992 the *TUGboat* editor put together a "Resource Directory". It was well received and many people looked forward to the next edition. But for lack of time (it took more than 100 hours to put together the 1992 edition) and resources (no volunteers were found to carry through this initiative) we never got around to publishing a new Directory. Anyway, nowadays it probably only makes sense to put such information on a Web site.

I hope we have answered most of your questions. We really appreciate your valuable input and suggestions. We can only hope that some of our other readers will also take the time to write to us, to tell us what they think, so that we can improve upon what we do for the benefit of all.

Since you expect to attend the TUG'97 conference this summer, we look forward to meeting you there.

Barbara Beeton and
Michel Goossens

Editorial Comments

Barbara Beeton

TEX Live!

With this issue of *TUGboat* you will find something unusual — the TEX Live CD-ROM. Sebastian Rahtz, Michel Goossens, and assorted others have been working hard over the past several months to assemble this production, and we are very pleased to present it here for all members to use and enjoy.

Briefly, this CD-ROM was prepared with Unix systems as the main target, but it also contains full packages of several other "public" TEX implementations and utilities — for Macintosh, MS-DOS, OS2 and Windows. The organization follows the standard TDS (TEX Directory Structure). On a Unix machine, TEX can be run either directly from the CD-ROM or installed on a hard disk.

The documentation for the CD-ROM appears on contrasting paper in this issue, so it should be easy to find. The full text is also included as file on the CD-ROM. (We really don't like to suggest that readers tear apart their issues.)

A TTN revival

At long last, some material intended for an abandoned issue of *TEX and TUG News* has been repackaged for *TUGboat*, and appears in this issue. One of these items is "Hey — it works!", a compendium of brief solutions to specific problems that can often be used "right out of the box".

We've been in communication with several of the former *TTN* columnists, and hope to have some *regular* columns starting with this year's December issue. One column we're pretty sure will return is "Typographer's Inn" (always one of my favorites); Peter Flynn is hard at work on a book about SGML Tools — like the rest of us, he is stretching his deadlines, but he will be done soon, and has promised to write up some new installments for us when he is.

New SGML extensions — watch for these

By its own description,

> Mathematical Markup Language, or MathML, is an XML application for describing mathematical expression structure and content. The goal of MathML is to enable mathematics to be served, received, and processed on the Web, just as HTML has enabled this functionality for text.

The full draft proposal, as presented to the World Wide Web Consortium (W3C), can be seen at the URL `http://www.w3.org/pub/WWW/TR/WD-math/`.

XML, the eXtensible Markup Language, is a dialect of SGML (the Standard Generalized Markup Language) intended for use on the Web as an alternative to HTML (HyperText Markup Language). From the draft XML specification:

> The goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML.
>
> For this reason, XML has been designed for ease of implementation, and for interoperability with both SGML and HTML.

An on-line FAQ (list of Frequently Asked Questions), maintained by Peter Flynn and others, can be found at `http://www.ucc.ie/xml/`.

Another kind of tugboat calendar

Take a look at the URL `http://www.halcyon.com/clcook/tugclndr.htm`. (This pointer should have been published last fall, but I mislaid the address.) It advertises a 1997 calendar illustrated with photos of historic and contemporary tugboats at work and play in the U.S. Pacific Northwest.

A copy of this calendar hangs on my office wall, where it reminds me of meetings and appointments, and also of the fact that not everyone works indoors slaving over a computer.

Tools for TEX users

Late last year, the Textures Reader was made freely available by Blue Sky Research. This can be used on any Macintosh-based system as a standalone previewer, or easily linked to Internet browsers as a helper application for automated network access to TEX DVI documents.

The file `Reader.sea.hqx` can be found on the Comprehensive TEX Archive Network (CTAN) in `systems/mac/textures/utilities/`, or from the `ftp.bluesky.com` or `www.bluesky.com` servers. Thanks to Barry Smith for releasing this useful tool.

The Type 1 implementations of the Computer Modern fonts previously distributed by Blue Sky Research and Y&Y, Inc., have been made freely available through the cooperation of a consortium of scientific publishers with the font developers. Members of the consortium include Elsevier Science, IBM Corporation, the Society for Industrial and Applied Mathematics (SIAM), Springer-Verlag, and the American Mathematical Society (AMS). In order to assure the authenticity of the CM/PS fonts, copyright will be held by AMS; however this is in no way meant to restrict the uses to which the fonts may be put.

The CM/PS fonts are available in Macintosh and PFB (binary Type 1) outline formats. The canonical version of the collection is located on the AMS FTP server, `e-math.ams.org`, in the area `/pub/tex/cmfonts/ps`, and mirrored onto CTAN in `fonts/cm/ps-type1/bluesky`. Thanks to Barry Smith, Berthold Horn, and the consortium members.

Finally, a reminder that the EDMAC manual, the documentation for the macro package of the same name, developed by John Lavagnino and Dominik Wujastyk for typesetting of critical editions, is available for sale from the TUG office. Send inquiries to `tug@mail.tug.org`.

Imprint — A new electronic newsletter

Imprint: The Newsletter of Digital Typography was announced in April, and the first few issues have now been distributed to subscribers via e-mail. The contents cover a variety of topics related to TEX, troff, SGML, PostScript, PDF, fonts, internationalization, etc., etc. For the present, *Imprint* will be issued monthly, but the schedule is subject to change.

The editor is Robert A. Kiesling, and his statement of purpose says, in part

> We would like this to be a forum for people (like myself) who handle every phase of document preparation from conception to final output.

Subscriptions can be entered by sending a human-readable request to the editor at `imprint@macline.com`.

Museums of printing and typography

An inquiry on the list `typo-l` a couple of months ago centered on the existence and locations of typography and printing museums in Europe. The responses included several museums I'd visited, as well as quite a few that I hadn't. It occurred to me that a list of such institutions would be a good addition to the TUG Web pages, a place to look if one is going on a trip and wants to find out what interesting places might be in the neighborhood.

We'll start up a Web page with the information we have now, and look forward to other contributions. Please identify the name of the museum or other institution, the city and address if you know it, and a brief description of what sort of holdings or exhibits are there, whether the hours are regular, or if special arrangements or permission are required. Anything related to typography, printing, papermaking, bookbinding, ... Maybe even stone carving: some very skilful type designers and calligraphers of the past couple centuries have also been stonecarvers, and one should remember that the classic form of the Roman alphabet is modeled after the inscriptions on Trajan's column in Rome.

Send contributions to `TUGboat@mail.tug.org`. And visit the TUG Web pages at `http://www.tug.org/`.

Thanks to DANTE

The German TEX group — Deutschsprachige Anwendervereinigung TEX e.V., at their last general meeting, appointed me "member of honour in recognition of [my] work for TEX and the TEX community."

The letter announcing this honor was accompanied by a beautiful book on typography, *Lesetypographie*, by Hans Peter Willberg and Friedrich Forssman.[1] This book, chock full of examples (good and bad), suggestions, analysis and guidance, has already provided much food for thought, and actual solutions to several problems.

I am delighted with this gift, and thank everyone associated with Dante for the honor.

◇ Barbara Beeton
American Mathematical Society
P.O. Box 6248
Providence, RI 02940 USA
`bnb@ams.org`

[1] Mainz: Verlag Hermann Schmidt, 1997. ISBN 3-87439-375-5

Final TTN Editorial

[Editor's note: Following the decision to merge *TEX and TUG News* (*TTN*) into *TUGboat*, a number of other items required publication, and so it is only as of this issue of *TUGboat* that the merger takes shape. A number of items originally destined for *TTN* now appear here, but only a few suffer from datedness. In his unpublished editorial regarding the merger, *TTN*'s editor, Peter Flynn, made a number of points which are best included in this return of *TTN*, now an integral component to *TUGboat*.

It is with a mixture of relief and delight that we include some of the material which was intended for publication in *TTN*, issues 4,3 and 4,4.]

TEX and TUG News and *TUGboat*

At the 1995 meeting in St. Petersburg Beach, it was decided that *TUGboat* and *TTN* should once again be published between the same covers.

The most overriding issue in the merger was that having two separate publications was diluting the effort and militating against members finding the information they want. Another factor was that it was considered less economically viable for TUG to maintain two publications, each with their own production budget.

Neither *TUGboat* nor *TTN* will change their own unique flavour, at least not immediately, and there is no question of either publication being forced to adopt any elements of style or editorial policy from the other (I have known Barbara Beeton and admired her work on *TUGboat* for too long to have any wish to impose on her). Barbara and I have also discussed the technical problems that will face us, and we feel sanguine that they can be overcome.

Nam et ipsa scientia potestas est

One *canard* I wish to lay to rest before we go any further, however. Much concern has been expressed by some scientific and technical members of TUG — TEX's original constituency — that *TTN* and the decision in Santa Barbara to unbundle *TUGboat* from the TUG membership fee meant that TUG and *TUGboat* are in some way being watered down.

To many members this will seem strange, as no-one who reads *TUGboat* or knows its reputation could for more than a second imagine that anyone associated with it would permit that to happen.

To other members, however, the *implication* that TUG should restrict *TUGboat* to technical discussions, and to mathematical or scientific work has unsettled them. Many non-scientist, non-technical TUG members use TEX for purposes *other* than for typesetting mathematics, a fact which has long been recognised by TUG, in that *TUGboat* regularly publishes articles on linguistic and other areas of the humanities (albeit necessarily of a technical nature).

The *raison d'être* of *TTN* was that TUG should try to embrace these members (and the potential membership in their hinterland) by providing a publication which did not assume as high a level of technical expertise as that possessed by some readers of *TUGboat*.

In this, *TTN* has, I believe, been uniquely successful. It has been harshly criticised for dragging TEX down to a non-technical level — and I am delighted to have been a part of that. I believe that only by providing this kind of service can we help those members (who wish to expand their use of TEX) start the technical learning curve.

It is therefore significant that in reviewing the publications of TUG, *TTN* should be brought within the fold of *TUGboat*. An increasing sophistication on the part of the non-technical users may indicate that perhaps four years of separate publication has had the desired effect.

◇ Peter Flynn
Computer Center
University College
Cork, Ireland
pflynn@curia.ucc.ie

Hints and Tricks

'Hey — It Works!'

Jeremy Gibbons

[Editor's note: Welcome once again to *"Hey — it works!"*, a column devoted to (LA)TEX tips and tricks. This was intended for the last (alas, never published) issue of *TTN*, and Jeremy should not be held directly responsible for it at this point in time. We hope to include this column regularly in *TUGboat*.]

– – * – –

Jonathan Fine of Cambridge, UK wrote in response to Allan Reese's article in *TTN* 4,2. Allan explained why inserting `\lowercase` into an `\equal` test, as in

```
\renewcommand{\NL}[1]
  {\def\thisletter{#1}%
   \ifthenelse{\equal{%
    \lowercase{\thisletter}}%
        {\lowercase{\thatletter}}}%
     {\\}%
     {\\[\smallskipamount]%
      \global\def\thatletter{#1}}%
   #1}%
```

did not produce a case-insensitive string comparison — the `\lowercase` happens too soon, even with judicious uses of `\expandafter`.

Jonathan observed that the problem can be solved by applying the `\lowercase` around the *definitions*, rather than the uses, of `\thisletter` and `\thatletter`:

```
\renewcommand{\NL}[1]
  {\lowercase{\def\thisletter{#1}}%
   \ifthenelse{\equal{\thisletter}%
    {\thatletter}}%
     {\\}%
     {\\[\smallskipamount]%
      \lowercase{\global\def\thatletter{#1}}}%
   #1}%
```

We also have three new items this issue. The first, is about typesetting long division problems; Barbara Beeton submitted the idea and the crucial part of its solution to me a couple of years ago, and I've rewritten it in LATEX. I've also shown Donald Arseneau's amazing macros for doing the whole long division problem automatically, given just the dividend and divisor. The second article, by Christine Thiele, shows how to produce two small separately-numbered figures side by side; I used it in my column in *TTN* 4,1. The final article, by Dennis Kletzing, is a followup to his presentation at the TUG Annual Meeting in Florida in July 1995; he shows here how to typeset automatically many short items of differing sizes in a grid layout. Enjoy!

– – * – –

Long division

Barbara Beeton
American Mathematical Society
bnb@ams.org

Donald Arseneau
Tri-University Meson Facility
asnd@erich.triumf.ca

Barbara Beeton was asked by a secretary for help with typesetting a long division problem, such as

$$\begin{array}{r} 949 \\ 13\overline{)12345} \\ 117 \\ \hline 64 \\ 52 \\ \hline 125 \\ 117 \\ \hline 8 \end{array}$$

Barbara solved the secretary's problem, and was particularly proud of the use of the parenthesis. I've taken her idea and rephrased it in terms of LATEX's `tabular` environment (I'm afraid Barbara is a die-hard `plain` TEXie!), which makes placing the digits relatively straightforward.

The long division above was produced by

```
\newdimen\digitwidth
    \settowidth\digitwidth{0}
\def~{\hspace{\digitwidth}}
\def\divrule#1#2{%
  \noalign{\moveright#1\digitwidth%
      \vbox{\hrule width#2\digitwidth}}}
13\,\begin{tabular}[b]{@{}r@{}}
      949 \\ \hline
      \big)\begin{tabular}[t]{@{}l@{}}
        12345 \\
        117  \\ \divrule{0}{4}
        ~~64 \\
        ~~52 \\ \divrule{2}{3}
        ~~125 \\
        ~~117 \\ \divrule{2}{3}
        ~~~~8
   \end{tabular}
\end{tabular}
```

The macro `\divrule#1#2` produces a rule the same width as `#2` digits, indented by the width of `#1` digits. (In most fonts, all digits have the same width.) Notice the `\big)` between the divisor and the dividend, and also the use of `~` as an active character, producing a space the size of a digit; this should all be done within a group, in order that the redefinition of `~` is not global. Notice also the use of

nested `tabular`s, one bottom-aligned and one top-aligned, for placing the various parts correctly.

Of course, TEX is quite capable of doing long division itself. Donald Arseneau has posted the following macros on `comp.text.tex`:

```
\newcount\gpten % power-of-ten, tells which
                % digit we're doing
\newcount\rtot
     % running total -- remainder so far
\newcount\scratch

\def\longdiv#1#2{%
     % long division: #1/#2;  integers only
 \vtop{\offinterlineskip
   \setbox\strutbox\hbox{%
   \vrule height 2.1ex depth .5ex width0ex}%
   \def\showdig{$\underline{%
    \the\scratch\strut}$\cr\the\rtot\strut\cr
       \noalign{\kern-.2ex}}%
   \global\rtot=#1\relax
   \count0=\rtot\divide\count0by#2%
        \edef\quotient{\the\count0}%
%
   % make list macro out of digits in quotient:
   \def\temp##1{\ifx##1\temp\else
     \noexpand\dodig ##1%
        \expandafter\temp\fi}%
   \edef\routine{\expandafter%
        \temp\quotient\temp}%
%
   % process list to give power-of-ten:
   \def\dodig##1{%
      \global\multiply\gpten by10\relax}%
   \global\gpten=1\relax\routine
    % to display effect of one digit
    % in quotient (zero ignored):
   \def\dodig##1{%
      \global\divide\gpten by10\relax
      \scratch =\gpten
      \multiply\scratch  by##1\relax
      \multiply\scratch  by#2\relax
      \global\advance\rtot-\scratch \relax
      \ifnum\scratch>0 \showdig \fi
      % must hide \cr in a macro to skip it
   }%
   \tabskip=0pt
   \halign{\hfil##\cr % \halign for entire
                      % division problem
     $\quotient$\strut\cr
     #2$\,\overline{\vphantom{\big)}}%
     \smash{\raise3.5\fontdimen8%
        \textfont3\hbox{$\big)$}}%
     \mkern2mu \the\rtot}$%
         \cr\noalign{\kern-.2ex}
     \routine \cr
        % do each digit in quotient
}}}
```

Given these macros, the long division

$$\begin{array}{r} 949 \\ 13\overline{)\,12345} \\ 11700 \\ \hline 645 \\ 520 \\ \hline 125 \\ 117 \\ \hline 8 \end{array}$$

is produced simply by typing

```
\mbox{\longdiv{12345}{13}}.
```

– – * – –

Side-by-side figures

Christina Thiele
Ottawa, Canada
`cthiele@ccs.carleton.ca`

Something I recently had to do in LATEX was provide two figures side-by-side. There are actually two types of situation: one is where the figures are related, and numbered 1a and 1b, for example; the other is where they are numbered separately, as 1 and 2. The first instance, sub-figures, can be handled by getting `subfigure.sty`, by Steven Douglas Cochran, from the nearest CTAN site. The second instance can be done with regular LATEX commands. I have to thank Barbara Beeton who forwarded mail from `comp.text.tex` from January 1994 when the subject arose there; also thanks to the exchanges posted by Andrew Justin Caird, Gabriel Zachmann and Tim Murphy (who pointed the way to `subfigure.sty`). Gabriel has recent informed me that another, more elegant, solution would be to use David Carlisle's `tabularx.dtx`, obtainable from CTAN in `tex-archive/macros/latex/packages/tools/`.

The bare-bones template I extracted from that correspondence was as follows (the references were to photos of Cree leggings, in an article on same):

```
\begin{figure}
  \begin{minipage}{5cm}
    \vspace{7cm}
    \caption{Woman's leggings}
  \end{minipage}
\hfill
  \begin{minipage}{5cm}
    \vspace{7cm}
    \caption{Man's leggings}
  \end{minipage}
\end{figure}
```

I sent this notion off as a possible item for *TTN*. But before I had a chance to get this simple version written up, a new application was found, in Charles Wells' piece on "Cross references in the bibliography" (*TTN* 4,1:7). There, the code to produce the

comparison between original and modified BibTeX put the `minipages` inside a `tabular`, making for an easy way to do two-column work without doing line-by-line two-columning. The `minipage` environment might be worth having an extended tutorial on its uses; I suspect there's a lot can be done with it.

– – * – –

Enumerated arrays

Dennis Kletzing
Stetson University, Florida
kletzing@bliss.stetson.edu

Typesetting the solutions manual for a book frequently involves creating an enumerated list involving many short answers. These answers are usually enumerated across a row. The simple way to do this is to start typing, with the result that the enumeration counter is not aligned vertically from one row to the next. Moreover, the ends of lines frequently wrap to the next line and do not look good. Surely there must be a better way to do this!

I started thinking abut this after I saw a message posted on `comp.text.tex` from someone asking what the secret was for writing macros. The person responding mentioned several things and concluded with the words "think boxes". If we "think boxes", it is not difficult to put together a macro that typesets enumerated arrays.

Basically, each item will be typeset in an `\hbox` which consists of three boxes; one for the label, one for the label separation, and one for the item itself. We first choose a basic unit length `\mitemwidth` to measure the length of items, a label width, and a label separation width. Extend the `\mitemwidth` to `\mtotalwidth`, which includes the label width and label separation. Finally, create a box, `\mitembox`, to hold the item to be typeset.

```
\newdimen\mitemwidth
    \mitemwidth=.1\textwidth
\newdimen\mitemlabelwidth
    \mitemlabelwidth=2em
\newdimen\mitemlabelsep
    \mitemlabelsep=0.5em
\newcount\mitemtempcount
\newcounter{mitemcounter}
\newdimen\mtotalwidth
     \mtotalwidth=\mitemwidth
\advance\mtotalwidth
     by\mitemlabelwidth
\advance\mtotalwidth
     by\mitemlabelsep
\newbox\mitembox
```

To ensure that each entry is typeset consistently, we use this rule: measure the width of the entry, including label and separation. Then typeset it in a box whose width is the next largest multiple of the unit length. Thus, if an entry requires 2.3 inches and the unit length is 1 inch, 3 unit lengths will be required. Here is the macro:

```
\def\mitem#1{%
     \refstepcounter{mitemcounter}%
  \setbox\mitembox=\hbox{%
    \hbox to \mitemlabelwidth{%
      \hfil\arabic{mitemcounter}.}%
    \hskip\mitemlabelsep\hbox{#1}%
       \hskip0pt}%
  \mitemtempcount=\wd\mitembox%
    \advance\mitemtempcount
        by\mtotalwidth%
    \advance\mitemtempcount by -1%
    \divide\mitemtempcount
        by \mtotalwidth%
  \setbox\mitembox=\hbox
        to\mitemtempcount\mtotalwidth{%
    \box\mitembox\hfill}%
  \leavevmode\box\mitembox\hskip0pt}
```

[Notice the idiom $(x + y - 1) \div y$ for rounding-up integer division of x by y, in terms of rounding-down division $\div$; also, the dimension `\mtotalwidth` is being used as a `\count`. –jg]

For example:

1. $x = 3$
2. $y = -2$
3. $x = 7$
4. 8
5. circle
6. $y = 2x^2$; parabola
7. $2x^2 + y^2 = 9$; ellipse
8. circle
9. $x^2 + y^2 = 9$
10. $x + y = 3$
11. line
12. $x = 21$
13. $y = 5$
14. $z = -5$
15. $x = 7$
16. $y = -9$
17. 2
18. 5
19. 6
20. -1
21. 9

This was set using the code

```
\begin{flushleft}
\mitem{$x=3$}
\mitem{$y=-2$}
\mitem{$x=7$}
\mitem{8}
\mitem{circle}
\mitem{$y=2x^2$; parabola}
\mitem{$2x^2+y^2=9$; ellipse}
\mitem{circle}
...
\end{flushleft}
```

In this example the `\mitemwidth` is set to `0.1\textwidth`. The user should experiment with different values of `\mitemwidth` to see how the shape of the array changes. This macro has the advantage that if changes are made to the entries, all items are renumbered and arranged appropriately. One disadvantage is that if an entry takes more than a single line, it will not wrap the line. Also, there are

some situations when we wish to wrap several lines within the entry. For these cases, there is a more elaborate package called `multienum.sty` which sets each item in a `\parbox` of width 1, 1/2, 1/3, or 1/4 of the `\textwidth`, wrapping the entry if necessary. Interested readers may contact me for a copy of it. Finally, I want to say a special word of thanks to Jeremy Gibbons for his help in putting together this version of the multienumerate macro.

◇ Jeremy Gibbons
School of Computing and
Mathematical Sciences
Oxford Brookes University
Gipsy Lane, Headington
Oxford, OX3 0BP UK
jgibbons@brookes.ac.uk

Humanities

TEX and the Humanities

Christina Thiele

Just a reminder to everyone working in the field of Humanities typesetting that we're always looking for new information, updates, and of course, elaborations on short notices which appear in this column.

To date, we've been able to bring you a revised version of a paper which had originally appeared in the Dutch group's journal: "A Medieval Icelandic manuscript", by Andrea de Leeuw van Weenen.[1] In future *TUGboat* issues, we hope to bring more news on critical editions; I myself am currently working on another John Donne crit-ed: *Essays in Divinity*, edited by Anthony Raspa of Laval University (Canada).

In response to the initial column, I've had mail from Jack Lincoln about some books he and Nigel Wiseman have set with CTEX (for Chinese characters), and about a work in progress, a clinical dictionary of Chinese medical terminology; Hans Hagen wrote that they've been using TEX for 8 years at Pragma (a publishing house in the Netherlands), and have developed an extensive package called ConTEXt (hope to get more on that later); and Eric Manning at the U.S. Dept. of Commerce also wrote with information on economics materials being typeset with LATEX (there'll definitely be more on that in the next column!). And finally, Robert Thomas from Manitoba (Canada) informs me that Series III of *Philosophia Mathematica* has been produced in plain TEX since 1993.

So, good news from all over!

– – * – –

The initial column[2] dealt with the areas of Collected works, Critical editions, Dictionaries, Journals, Linguistics — not to mention that perennial grab-bag, "Other projects". The next column, in addition to providing updates on the above, will look at the following topics: Music, Economics, and Bibles. At least, that's what we have so far. If there are other topics you'd like to see included, or for which you have interesting information, please send it along!

To try and keep the column to a reasonable length, only publications dated from 1994 onwards are included (with a few exceptions). I'm still hoping to eventually have these publications listed on the Web, which would allow not only more complete entries and information about each book's TEX-ing particulars, but also would make it possible to have an archival place of record for everything that's gone before. If anyone's good at doing bibliographic work (ideally, in BIBTEX format; even more ideally, in sync with that used in Nelson Beebe's collection), please get in touch!

Suggestions on how to make this a lively place to turn to when your *TUGboat* issue arrives are always welcome and needed. If information on your project was not quite correct, let us know so that a correction note can be printed.

At this stage, we're just hungry for information, news, book details, and so on. Offers of help with the bibliographic work would definitely be appreciated: keeping track of what's been submitted, making sure entries are as complete as possible, and eventually getting them into shape for the Web — that sort of thing.

Send mail to `cthiele@ccs.carleton.ca`, and use the subject line `TB-Humanities`, to help sort things out quickly.

◇ Christina Thiele
15 Wiltshire Circle
Nepean, Ontario
K2J 4K9 Canada
`cthiele@ccs.carleton.ca`

[1] "Diplomatic edition of a medieval Icelandic manuscript", *MAPs* 14 (95.1), pages 31–34; *TUGboat* 18.1 (1997), pages 30–36.

[2] *TUGboat* 17.4 (1997), pages 388–393.

Queries

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.

– – * – –

Volunteers Needed: PSfrag hackers

Hey folks,

If you haven't used it before, PSfrag is a cool little package first created by Craig Barratt (formerly of Stanford University) to mix Encapsulated PostScript figures with LaTeX/TeX constructions. Basically it lets you replace pieces of text in an EPS figure with an equation or some other typeset text. That way, you can use your favorite drawing package to build your figures without having to lament its lack of support for scientific text or equations.

It supports LaTeX 2$_\varepsilon$ using DVIPS or DVIPSone from Y&Y to handle the DVI to PostScript conversion. (And with a little help from CTAN, it even supports Plain TeX.) Psfrag is available on CTAN (`/archive/macros/latex/contrib/supported/psfrag`), and is included with teTeX. It's fairly well documented, too (see `texmf/doc/latex/psfrag` in the teTeX tree, for example.)

I took it over from Craig several years ago, and incorporated some improvements provided by the TeX community. More recently I gave it a complete overhaul to make it more portable and easier to use. In fact, if you haven't used it in a couple of years, you should get the latest version and try it.

Unfortunately, I'm finishing my career as a graduate student very soon, and I'm not going to have the freedom to work on it like I used to. I will continue to support PSfrag with bug fixes as time permits, but I'll probably be slow — and I certainly will not be adding any new features. (Perhaps this is a good thing!) Eventually I may even have to stop supporting PSfrag altogether — after all, if I stop *using* it (gasp!), I may no longer remain qualified to support it!

So I'd like to hear from volunteers to take over the reins to support PSfrag. Maintaining PSfrag does require some skill in both TeX macros and the PostScript language — some of the macros are pretty tricky! You can certainly count on us veterans for help, of course. Ideally, I'd like to see two or three people collaborate on PSfrag support, with one leading.

If you are interested:

1. Grab the latest version from CTAN, unpack it, and play with the code. See with you think! Do you think you can handle it?
2. Respond to the PSfrag maintainer's mailing list at `psfrag@rascals.stanford.edu` with your intentions. Obviously we don't want people working in the dark! Still, you will want to look at the code first to see if it suits you.

Thanks, everyone!

PSfrag mailing list: `psfrag@rascals.stanford.edu`

◇ Michael C. Grant, S.T.B.Ph.D.
(soon to be Ph.D.)
`mcgrant@numeritech.com`

– – * – –

Features for a WYSIWYG LaTeX editor

I am a University student in Australia and I am working on my thesis topic *A WYSIWYG Editor for LaTeX*. Therefore, I would like to gather some information from LaTeX users around the world on what they think of LaTeX and what they expect from a WYSIWYG LaTeX editor.

Survey

1. What do you expect from a WYSIWYG LaTeX Editor (e.g. any features you would like such as undo, or other features specific to LaTeX?) Please list as many as you can think of.
2. What LaTeX system (such as MikTeX for Windows) do you currently use? And what platform do you use LaTeX on?
3. Do you use any of the current WYSIWYG LaTeX editors (such as WysiTeX) and if so, please name them.
4. If your answer to 2) is *yes*, could you list some of the shortcomings in the existing packages and what would you suggest to improve on those packages?
5. Why would you use LaTeX instead of other text formatting applications such as Microsoft Word for Windows?
6. Would you rather have ... (pick one of two)
   Full WYSIWYG editing (i.e., Built-in display) or "Code input option + preview" (External previewer)

If possible, please forward your reply to my email address, below. Thanks!

◇ Sidney Chow
`sidneyc@cse.unsw.edu.au`

Publications

[Editor's note: Most of the following publications were to have appeared in the *TTN* 4,3 issue in December of 1995, under the by-line of Peter Schmitt (who is in no way responsible for the obvious delay). We hope that this information will be of use to our readers, and should you come across publications which you think would be of interest to the TEX community, please let us know by sending email to `tugboat@mail.tug.org`.]

Books

George Grätzer: *Math into LaTeX An Introduction to LaTeX 2ε and AMS-LaTeX 1.2.* Boston-Basel-Berlin: Birkhäuser, 1995.
Approx. 442pp., 8 illus. Ca. $42.50. ISBN0-8176-3805-9.

• This book extends the scope of the author's *Math into TEX* (on AMS-LaTeX) to the newest version of both LaTeX and AMS-LaTeX while retaining its introductory nature. It will both help the novice user to get started and the experienced user to extend the range of applications. [adapted from promotional material]

David Slaomon: *The Advanced TEXbook.* Berlin-Heidelberg-London: Springer-Verlag, 1995.
xx, 482pp. Ca. $39.95/DM 68. ISBN0-387-94556-3.

• "TEX is in essence a programming language and so it is best viewed from this perspective. In this book, the author presents a complete course in TEX which will be suitable for users of TEX who want to advance beyond the basics. Throughout, numerous examples are given and exercises (with answers) provide a means for readers to test their understanding of the material." [adapted from promotional material]

Raymond Seroul & Silvio Levy: *A Beginner's Book of TEX.* Springer, 1995. 1st ed. 1991. Corr. 3rd printing 1995. xii, 284pp. DM 61. ISBN3-540-97562-4.

Helmut Kopka and Patrick W. Daly: *A Guide to LaTeX 2ε: Document Preparation for Beginners and Advanced Users*, Second edition. Addison Wesley Reading, MA, USA, 1995. x, 554pp., Ca. $37.61, ISBN0-201-42777-X.

• If you are a user with little or no experience of computers or text formatting and you want to master LaTeX to produce documents of high quality, this is the book for you. Fully revised to cover both LaTeX 2.09 and the latest version of LaTeX2e, this tutorial contains an exciting new text design that makes it even more accessible than before. [From the back cover of the book]

Petr Olšák: *Typografický systém TEX.* CsTEX Users Group, Praha, Czech Republic, 1995. ISBN80-901950-0-8, 272pp.

Jiři Rybička: *LaTeXpro začátečníky.* Brno, Czech Republic, Konvoy, 1995. ISBN80-85615-42-8.

David Salomon: *The Advanced TEXbook.* Berlin, Germany / Heidelberg, Germany / London, UK / etc., Springer-Verlag, 1995. xx, 490pp., Ca. $39.95, ISBN0-387-94556-3.

George Grätzer: *Math into LaTeX: an introduction to LaTeX 2ε and AMS-LaTeX 1.2.* Basel, Switzerland, Birkhäuser, 1996. xxvii, 451pp., Ca. $49.50, ISBN0-8176-3805-9, 3-7643-3805-9.

Helmut Kopka: *LaTeX Band 3: Eweriterungen*, Bonn, Germany, Addison-Wesley Verlag, 1997. Ca. DM69.90, ISBN3-89319-666-8.

Michel Goossens, Sebastian Rahtz and Frank Mittelbach: *The LaTeX Graphics Companion. Illustrating documents with TEX and PostScript.* Reading, MA: Addison Wesley Longman, Inc., 1997. 555 pp. ISBN0-201-85469-4.

• This book inaugurates a new and broad-ranging series, entitled "Addison-Wesley Series on Tools and Techniques for Computer Typesetting". Future titles will address the needs both of LaTeX users and of LaTeX developers, and will also cover other systems of value to the scientific and technical community in the preparation of high-quality typeset documents. This handy reference describes techniques and tricks needed to illustrate LaTeX documents, and answers common user questions about graphics and PostScript fonts. It provides the first full description of the standard LaTeX color and graphics packages, and shows how you can combine TEX and PostScript capabilities to produce beautifully illustrated pages. [From the book.]

Electronic Publications

Tobias Oetiker: *The not so Short Introduction to LaTeX 2ε. Version 2.0.*
June 1995. 80pp. Available from CTAN (—info/lshort—).

• "This document teaches you all you need to know in order to typeset reports, articles, books and slides in LaTeX 2ε." [from the author's announcement]

TEX Live CD-ROM

The TEX Live Guide, version 2

Sebastian Rahtz and Michel Goossens

Contents

– – * – –

1 Introduction

This documentation describes the main features of the **TEX Live** CD-ROM, a TEX/LATEX distribution for Unix, Windows32, Amiga and NeXT systems, that includes TEX, LATEX 2ε, METAFONT, MetaPost, many other programs such as Makeindex, dvips, xdvi and BIBTEX; and a very complete set of macros, fonts and documentation conforming to the *TEX Directory Standard* which can be used with nearly every TEX setup.

This TEX package uses the Web2c implementation of the programs, which tries to make TEXing as easy as possible, and takes full advantage of the efficient and highly customizable Kpathsea library from Karl Berry. It can be run either directly from the CD-ROM, or installed on a hard disk.

The **TEX Live** runnable systems contain two experimental extensions to normal TEX:

1. ε-TEX, which adds a small but powerful set of new primitives, and the TEX--XET extensions for left to right typesetting; in default mode, ε-TEX is 100% compatible with ordinary TEX. See `share/texmf/doc/html/e-tex/etex.htm` on the CD-ROM for details.
2. pdfTEX, which can optionally write Acrobat PDF format instead of `dvi`; there is no formal documentation for this yet, but the file `share/texmf/tex/pdftex/example.tex` shows how it is used. The LATEX `hyperref` package has an option 'pdftex' which turns on all the program features.

While ε-TEX is stable, pdfTEX is under continual development; the version on the CD-ROM may not be stable. Most platforms have version 0.11 of May 7th, but some have a slightly earlier one of May 5th, which may have problems including PNG files.

The entire GUTenberg distribution for Windows is included on the CD-ROM, ready to install, as are the following complete packages:

- OzTEX 3.0 for Macintosh
- CMacTeX 2.6 for Macintosh
- Macintosh utilities (Alpha, Excalibur, etc.)
- MikTeX for Windows 95
- emTeX for MSDOS and OS/2
- TEX shells for Windows and DOS (Winedt, e4t, TeXtelmExtel, emTeXgi)

These are provided unchanged from CTAN, and have not been integrated in any way with the rest of the CD-ROM.

1.1 History and acknowledgements

This CD-ROM distribution is a joint effort by the TEX Users Group, the UK TEX Users Group, and the French TEX Users (GUTenberg), with the support of the Dutch, German and Czech/Slovak user groups. Discussion began in late 1993 when the Dutch TEX Users Group was starting work on its 4AllTEX CD-ROM for MSDOS users, and it was hoped at that time to issue a single, rational, CD-ROM for all systems. This was far too ambitious a target, but it did spawn not only the very successful 4AllTEX CD-ROM, but also the TUG Technical Council working group on a *TEX Directory Structure*, which specified how to create consistent and manageable collections of TEX support files. The final draft of the TDS was published in the

December 1995 issue of *TUGboat*, and it was clear from an early stage that one desirable product would be a model structure on CD-ROM. The CD-ROM you now have is a very direct result of the working group's deliberations. It was also clear that the success of the 4AllTEX CD-ROM meant that Unix users would benefit from a similarly easy system, and this is the other main strand of **TEX Live**.

We undertook to make a new Unix-based TDS CD-ROM in the autumn of 1995, and quickly identified Thomas Esser's teTEX as the ideal setup, as it already had multi-platform support and was built with portability across file systems in mind. Thomas agreed to help, and work began seriously at the start of 1996. The first edition was released in May 1996. At the start of 1997, Karl Berry completed a major new release of his Web2c package, which included nearly all the features which Thomas Esser had added in teTEX, and we decided to base the 2nd edition of the CD-ROM on the standard Web2c, with the addition of teTEX's `texconfig` script.

We are particularly grateful to: Karl Berry for extra advice, encouragement, and (of course) for providing the Web2c distribution; Thomas Esser, without whose marvellous teTEX package this CD-ROM would certainly not exist, and whose continual help makes it a better product; and Ulrik Vieth, for checking many assumptions at the start, and providing a great deal of extra material for the documentation tree.

Fabrice Popineau did the excellent port of Web2c 7.0 to Windows 95/NT and provided much help; Andreas Scherer contributed the Amiga compilation; Gregor Hoffleit contributed the TeXview material for NextStep users, and the NextStep binaries. At Florida State University Supercomputer Research Institute, Mimi Burbank arranged access to a slew of different computers to compile TEX on, and acted as an essential guinea-pig whenever asked. Michel Goossens provided access to computers at CERN, and Robin Fairbairns stepped in to provide an Alpha running Linux at Cambridge.

Some of this documentation is drawn from the teTEX guide by Thomas Esser and Dirk Hillbrecht; the catalogue of packages depends very much on the ongoing work of Graham Williams (`mailto:Graham.Williams@cbr.dit.csiro.au`), who kindly agreed to allow us to use it here. Mimi Burbank, Robin Fairbairns and Ulrik Vieth worked hard to improve this text.

1.2 Future versions

This CD-ROM is not a perfect product! We plan to re-issue it once a year, and would like to provide more help material, more utilities, more installation programs, and (of course) an ever-improved and checked tree of macros and fonts. This work is all done by hard-pressed volunteers in their limited spare time, and a great deal remains to be done. If you can help, don't hesitate to put your name forward!

Corrections, suggestions and additions for future revisions should be sent to:

Sebastian Rahtz
7 Stratfield Road
Oxford OX2 7BG
United Kingdom
`mailto:s.rahtz@elsevier.co.uk`

Updates, notes, and suggestions will be made available on CTAN in `info/texlive`. A WWW page for information and ordering details is at `http://www.tug.org/tex-live.html`.

2 Structure and contents of the CD-ROM

The CD-ROM top level directories are:

bin The TEX family programs, arranged in separate platform directories;

info Documentation in GNU 'info' format for the TEX system;

macintosh The OzTEX and CMacTeX packages ready to install, plus some other utilities;

man Documentation in Unix man pages for the TEX system;

msdos DOS TEX packages — emTeX, and three TEX shells;

support The source of all programs, including the main Web2c TEX and METAFONT distribution; this directory also includes various bits of TEX-related software which are *not* installed by default, such as MusixTEX support programs, and a complete set of Ghostscript;

share The main support tree of macros, fonts and documentation;

wingut The GUTenberg distribution for Windows; this consists of compressed archives which must be unpacked and installed on a hard disk. Please see the detailed instructions in French;

win32 TEX packages for Windows 95 and NT users (MikTeX, and the original package of the Win32 port of Web2c).

There are also two installation scripts for Unix systems, `install-cd.sh` and `install-pkg.sh`; we discuss them on p. 83.

2.1 The TDS tree

The **TEX Live** `share/texmf` tree consists of various 'collections', each of which has a set of 'packages', of

which there are over 400 on the CD-ROM. Normal installation allows the user to copy all of a collection to a local hard disk from the CD-ROM, but it is also possible to install just one package of a collection. Each of the collections is divided into *basic* (1), *recommended* (2) and *other* (3). The collections are:

ams The American Mathematical Society macro packages and fonts

bibtex BibTeX styles and databases

doc General guides and documentation in various formats, including HTML and PDF

dvips Support for Rokicki's dvi to PostScript driver

fonts Font sources, metrics, PostScript and bitmap forms

formats Eplain, RevTeX, physxx, texsis, alatex, text1, lollipop, etc.

generic Extra macros for use with any format

graphics Macro packages for graphics

lang Support for non-English languages

latex LaTeX, including official tools and all LaTeX 2_ε contributed packages

metapost Support for MetaPost

plain Macros for plain TeX

systems Binaries for Unix platforms

texlive Basic material for the distribution

The appendix starting on p. 101 lists all the packages in alphabetical order with the collection they are found in, and a brief description. Thus all packages in collection `latex1` are what one must have to get started with LaTeX, packages in `latex2` are recommended for most users, and `latex3` contains optional packages. The directory `share/texmf/lists` contains lists of all files in each package (used by the installation package).

3 Installation and use under Unix

You can use the **TeX Live** CD-ROM in three ways:

1. You can mount the CD-ROM on your file system, adjust your `PATH`, and run everything off the CD-ROM; this takes very little disk space, and gives you immediate access to everything on the CD-ROM; although the performance will not be optimal, it is perfectly acceptable on, for instance, PCs running Linux;
2. You can install all or part of the system to your local hard disk; this is the best method for many people, if they have enough disk space to spare (a minimum of about 10 megabytes, or 100 megabytes for a recommended good-sized system);
3. You can install selected packages to work either with your existing TeX system or a **TeX Live** system you installed earlier.

Each of these methods is described in more detail in the following sections.

Warning: This CD-ROM is in ISO 9660 (High Sierra) format, with Rock Ridge extensions. In order to take full advantage of the CD-ROM on a Unix system, your system needs to be able to use the Rock Ridge extensions. Please consult the documentation for your *mount* command to see if it is possible. If you have several different machines on a local network, see if you can mount the CD-ROM on one which *does* support Rock Ridge, and use it from the others.

Linux, FreeBSD, Sun, SGI and DEC Alpha systems should be able to use the CD-ROM with no problems. We would appreciate receiving detailed advice from other system users who also succeed, for future versions of this documentation.

The discussion below about installation assumes you have been able to mount the CD-ROM with full Rock Ridge compatibility.

3.1 Running TeX Live from the CD-ROM

The organisation of Web2c means that you can run programs simply by adding the appropriate directory under `bin` on the CD-ROM to your `PATH`, and the support files will all be found with no further ado. The following table shows the list of available directories and the systems they apply to.

alpha-linux	DEC Alpha Linux
alpha-osf3.2	DEC Alpha OS 3.2
amiweb2c	Amiga
hppa11-hpux9.05	HP9000 HPUX 9.05
hppa11-hpux10.20	HP9000 HPUX 10.20
i386-linux	Intel PC with Linux (ELF)
i586-freebsd2.2	Intel PC with Free BSD
i686-linux	Intel Pentium Pro with Linux
mab-nextstep3	NextStep 3
mips-irix4.0.5	SGI IRIX 4.0.5
mips-irix5.3	SGI IRIX 5.3
mips-irix6.3	SGI IRIX 6.3
mips-ultrix4.4	DECstation Ultrix 4.4
rs6000-aix3.2.5	IBM RS 6000 AIX 3.2.5
rs6000-aix4.1.1	IBM RS 6000 AIX 4.1.1
sparc-sunos4.1.3	Sun Sparc Sunos 4.1.3
sparc-solaris2.5	Sun Sparc Solaris 2.5
sparc-solaris2.4	Sun Sparc Solaris 2.4
sparc-linux	Sun Sparc Linux
win32	Windows 95 or NT

You may worry that when you subsequently make fonts or change configuration, things will go wrong because you cannot change files on the CD-ROM. However, you can maintain a parallel, writeable, TEX tree on your hard disk; this is searched before the main tree on the CD-ROM. The default location is `/usr/local/texmf`, but you can override this by setting the `TEXMFLOCAL` environment variable.

Thus *sh* or *bash* users on an Intel PC running Linux who mount the **TEX Live** CD-ROM on `/cdrom` by issuing the command:

```
mount -t iso9660 /dev/cdrom /cdrom
```

might add the following to their `.profile` script:

```
PATH=/cdrom/bin/i386-linux:$PATH
export PATH
```

If in doubt, ask your local system support guru to help you work out how to mount your CD-ROM or which directory to use for your system.

Appropriate support files will be installed on your hard disk the first time you need them. It is a good idea to immediately run the `texconfig` script to initialize things, and check it all works.

3.2 Installing TEX Live to a hard disk

All of the necessary steps to install all or part of the distribution on your hard disk are achieved by mounting the CD-ROM, changing to the top-level directory, and typing:

```
sh install-cd.sh
```

(On some Unix systems, you may need to use `sh5` or `bsh`). This works by accessing lists of collections and packages from the CD-ROM, and trying to guess what sort of computer system you are on; it should start by displaying the following:

```
Initializing collections... Done.
Counting selected collections... Done.
Calculating disk space requirements for
                  collections...Done.
Initializing system packages... Done.
```

It will then show the main control screen (Figure 1), which lets you change four things:

1. the type of system you are on, or want to install for;
2. the collections you want to install, at *basic*, *recommended* or *other* level;
3. the location on your hard disk to put the files;
4. some runtime behaviour features.

You choose options by typing a letter or number and pressing return. In the example, a Linux ELF system has been detected, the default of all collections to recommended level has been chosen, and the default installation directory is `/usr/local`; note that the disk space required for the current installation configuration is also displayed. If you make a suggested setup, you need about 172 megabytes of disk free; however, the basic setup will only take about 10 megabytes, and you can enhance it with selected packages as you need them.

Under the directory you choose for installation, the installation script will put the binaries in a subdirectory of `bin`, and the support tree in `share/texmf`.

The `options` item lets you decide whether to make new fonts be created in another location (if you want the main package mounted read-only for most users), and whether to make symbolic links for the *man* and GNU *info* pages in the 'standard' locations.

When you choose `<C>` for collections, you will see the display of available collections, the level of installation selected, and the disk space required (Figure 2). You can set alternative levels of installation for each collection, ranging from *none* to *all*. You can either set this for all collections at once, or choose a particular collection and set its level (Figure 3).

When you are finished, return to the main screen, and ask the installation to start. It will take each of the collections and systems that you requested, consult the list of files on the CD-ROM, and build a master list of files to transfer. These will then be copied to your hard disk, and the initialization sequence run (creating format files etc.). When this has finished, all you need do is add the correct subdirectory of `bin` in the TEX installation to your path, and start using TEX. If you want to move the binaries up one level, e.g. from `/usr/local/bin/alpha-osf3.2` to `/usr/local/bin`, you need to edit `share/texmf/web2c/texmf.cnf` and change the line

```
prefix = $SELFAUTOPARENT
```

to

```
prefix = $SELFAUTODIR
```

You can of course change the value of `prefix` to any directory you like, and move the support directory there.

3.3 Installing individual packages from TEX Live to a hard disk

You may want to use the **TEX Live** CD-ROM to either update an existing setup, or add features to an earlier installation from the CD-ROM. The main installation program is intended for the first time only, and subsequently you should use the

```
==================> TeX Live installation procedure <==========
===> Note: Letters/digits in brackets indicate menu items <===
===>       for commands or configurable options           <===
    Detected system: Intel PC with Linux (ELF)
    <C> collections:    21 out of 30, disk space required: 163955 kB
    <S> systems:         1 out of 20, disk space required:   7946 kB
                               total disk space required: 171901 kB
    <D> directories:
          TEXDIR = /usr/local
    <O> options:
          [ ] alternate directory for automatically generated fonts
          [ ] create symlinks in standard directories
    Other commands:
          <I> start installation,  <H> help,  <Q> quit
Enter command:
```

Figure 1: Installation screen, example 1

```
Current collections setup:           total size : 171901 kB
============================================================
        name      selection         size
  <1>   ams       [recommended]     6359 kB
  <2>   bibtex    [recommended]     6584 kB
  <3>   doc       [recommended]    26531 kB
  <4>   dvips     [recommended]      563 kB
  <5>   fonts     [recommended]    21862 kB
  <6>   formats   [recommended]     1003 kB
  <7>   generic   [recommended]      501 kB
  <8>   graphics  [recommended]    10373 kB
  <9>   lang      [recommended]     3287 kB
  <W>   metapost  [recommended]     1280 kB
  <X>   latex     [recommended]    28333 kB
  <Y>   plain     [recommended]      756 kB
  <Z>   texlive   [recommended]    56523 kB
                            SUM:  163955 kB
================================================
  global commands: select <n>one / <b>asic / r<e>commended / <a>ll
                   for all collections
  <R>   return to platform menu
  <Q>   quit
Enter command to modify current selection:
```

Figure 2: Installation screen, example 2

```
Collection: Fonts
================================================================================
Fonts, including metrics, virtual fonts and sources
================================================================================
  <N>  no packages
  <B>  basic packages                 [  2007 kB]
  <E>  basic + recommended packages   [ 21862 kB]
  <A>  all packages                   [ 34303 kB]
================================================================================
  <R>   return to collection menu
  <Q>   quit
Enter command:
```

Figure 3: Installation screen, example 3

`install-pkg.sh` script on the CD-ROM. Run this by mounting the CD-ROM, changing to the mounted directory, and typing

`sh install-pkg.sh` *options*

The script supports nine options; the first four let you set the individual package you want to install, the whole collection (i.e., `ams2`), the name of the mounted CD-ROM directory, and the name of the directory containing the list files (normally these latter two will be set automatically):

`--package=`*name*
`--collection=`*name*
`--cddir=`*name*
`--listdir=`*name*

What actually happens is controlled by four more switches; the first two allow you to exclude documentation or source files from the installation; the third stops the default action of running `MakeTeXls-R` on completion to rebuild the file database, and the last does nothing but list the files that would be installed:

`--nodoc`
`--nosrc`
`--nohash`
`--listonly`

Finally, you can specify that instead of installing the files, the script should make a *tar* archive in a specified location:

`--archive=`*name*

Thus, if we simply wanted to see the files that make up the package `fancyhdr` before we installed, our command and output would be as follows:

```
sh install-pkg.sh --package=fancyhdr --listonly
texmf/doc/latex/fancyhdr/fancyhdr.dvi
texmf/doc/latex/fancyhdr/fancyhdr.tex
texmf/lists/latex3/fancyhdr
texmf/source/latex/fancyhdr/README
texmf/source/latex/fancyhdr/fancyheadings.new
texmf/tex/latex/fancyhdr/extramarks.sty
texmf/tex/latex/fancyhdr/fancyhdr.sty
texmf/tex/latex/fancyhdr/fixmarks.sty
```

Other examples of usage are:

- Install the LaTeX package `arseneau`:

  ```
  install-pkg.sh --package=arseneau
  ```

- Install the LaTeX package `alg` with no source files and no documentation:

  ```
  install-pkg.sh \
       --package=alg --nosrc --nodoc
  ```

- Install all the packages available in the 'extra' Plain TeX collection:

  ```
  install-pkg.sh --collection=plain3
  ```

- Place all files which are need for PSTricks in a tar file in `/tmp`:

  ```
  install-pkg.sh --package=pstricks \
         --archive=/tmp/pstricks.tar
  ```

3.4 `texconfig`

After the installation program has copied all files to their final locations, you can call a program called `texconfig` that allows you to configure the system to fit your local needs. This can be called at any other time to change your setup, with a full-screen (which requires the `dialog` program) or command-line interface. It should be used for all maintenance, like changes of installed printers, or rebuilding of the file database. Both modes have help text to guide you through the facilities.

3.5 Building on a new platform

If you have a platform for which we have not provided binary sources, you will need to compile TeX and friends from scratch. This is not as hard as it sounds. What you need is all in the directory `support/texk-7.0` on the CD-ROM.

To compile TeX, you should get gcc, flex and a recent version of GNU make. gcc-2.5.8, flex-2.4.7 and GNU make-3.72.1 or newer should be fine. You may be able to work with other C compilers and Make programs, but you will need a good understanding of building Unix programs to sort out problems.

You should first install the support tree from the **TeX Live** CD-ROM (do a basic install, with no system binaries chosen). Then copy the `texk-7.0` directory to your disk, and run

```
configure -prefix=$TEXMF
```

where `$TEXMF` is the place where you installed **TeX Live**.

Now type `make install-exec` and relax...

4 A user's guide to the Web2c system

Web2c contains a set of TeX-related programs, i.e., TeX itself, METAFONT, MetaPost, BibTeX, etc. The original implementation was by Tomas Rokicki, who in 1987 developed a first TeX-to-C system adapting change files under Unix, which were primarily the work of Howard Trickey and Pavel Curtis. Tim Morgan became the maintainer of the system, and during this period the name changed to Web-to-C. In 1990, Karl Berry took over the work, assisted by dozens of additional contributors. The latest result is Web2c Version 7, which was released in February 1997, and forms the basis of the present **TeX Live** CD-ROM.

The Web2c 7.0 system runs on Unix, Windows 95/NT, DOS, Amiga, and other operating systems. It uses Knuth's original sources for TeX and other

basic programs written in `web` and translates them into C source code. Moreover, the system offers a large set of macros and functions developed to augment the original TEX software. The most commonly used components are:

`bibtex` Maintaining bibliographies.
`dmp troff` to MPX (MetaPost pictures).
`dvicopy` Virtual font expansion.
`dvitomp` DVI to MPX (MetaPost pictures).
`dvitype` DVI to human-readable text.
`gftodvi` Generic font proofsheets.
`gftopk` Generic to packed fonts.
`gftype` GF to human-readable text.
`makempx` MetaPost label typesetting.
`mf` Creating typeface families.
`mft` Prettyprinting METAFONT source.
`mpost` Creating technical diagrams.
`mpto` MetaPost label extraction.
`newer` Compare modification times.
`patgen` Creating hyphenation patterns.
`pktogf` Packed to generic fonts.
`pktype` PK to human-readable text.
`pltotf` Property list to TFM.
`pooltype` Display WEB pool files.
`tangle` WEB to Pascal.
`tex` Typesetting.
`tftopl` TFM to property list.
`vftovp` Virtual font to virtual property list
`vptovf` Virtual property list to virtual font.
`weave` WEB to TEX.

The precise functions and syntax of these programs are described in the documentation of the individual packages or of Web2c itself. However, knowing a few principles governing the whole family of programs will help you to benefit optimally from your Web2c installation.

All programs honor the standard GNU options:

`-help` print basic usage summary.
`-verbose` print detailed progress report.
`-version` print version information, then exit.

For locating files the Web2c programs use the path searching library Kpathsea. This library uses a combination of environment variables and a few configuration files to optimize searching the TEX directory tree. Web2c 7.0 can handle more than one directory tree simultaneously, which is useful if one wants to maintain TEX's standard distribution and local extensions in two distinct trees. To speed up file searches the root of each tree has a file `ls-R`, containing an entry showing the name and relative pathname for all files "hanging" under that root.

4.1 `Kpathsea` path searching

Let us first describe the generic path searching mechanism of the Kpathsea library.

We call a *search path* a colon- or semicolon-separated list of *path elements*, which are basically directory names. A search path can come from (a combination of) many sources. To look up a file "`my_file`" along a path "`.:/dir`", Kpathsea checks each element of the path in turn: first `./my_file`, then `/dir/my_file`, returning the first match (or possibly all matches).

In order to adapt optimally to all operating systems' conventions, on non-Unix systems Kpathsea can use filename separators different from "colon" (":") and "slash" ("/").

To check a particular path element p, Kpathsea first checks if a prebuilt database (see "Filename database" on p. 89) applies to p, i.e., if the database is in a directory that is a prefix of p. If so, the path specification is matched against the contents of the database.

If the database does not exist, or does not apply to this path element, or contains no matches, the filesystem is searched (if this was not forbidden by a specification starting with "`!!`" and if the file being searched for must exist). Kpathsea constructs the list of directories that correspond to this path element, and then checks in each for the file being searched for.

The "file must exist" condition comes into play with VF files and input files read by TEX's `\openin` command. Such files may not exist (e.g., `cmr10.vf`), and so it would be wrong to search the disk for them. Therefore, if you fail to update `ls-R` when you install a new VF file, it will never be found.

Each path element is checked in turn: first the database, then the disk. If a match is found, the search stops and the result is returned.

Although the simplest and most common path element is a directory name, Kpathsea supports additional features in search paths: layered default values, environment variable names, config file values, users' home directories, and recursive subdirectory searching. Thus, we say that Kpathsea *expands* a path element, meaning transforming all the specifications into basic directory name or names. This is described in the following sections in the same order as it takes place.

Note that if the filename being searched for is absolute or explicitly relative, i.e., starts with "/" or "./" or "../", Kpathsea simply checks if that file exists.

4.1.1 Path sources

A search path can come from many sources. In the order in which Kpathsea uses them:

1. A user-set environment variable, for instance, `TEXINPUTS`. Environment variables with a period and a program name appended override; e.g., if "`latex`" is the name of the program being run, then `TEXINPUTS.latex` will override `TEXINPUTS`.
2. A program-specific configuration file, for example, a line "`S /a:/b`" in `dvips`' `config.ps`.
3. A Kpathsea configuration file `texmf.cnf`, containing a line like "`TEXINPUTS=/c:/d`" (see below).
4. The compile-time default.

You can see each of these values for a given search path by using the debugging options (see "Debugging actions" on p. 92).

4.1.2 Config files

Kpathsea reads *runtime configuration files* named `texmf.cnf` for search path and other definitions. The search path used to look for these files is named `TEXMFCNF` (by default such a file lives in the `share/texmf/web2c` subdirectory). *All* `texmf.cnf` files in the search path will be read and definitions in earlier files override those in later files. Thus, with a search path of "`.:$TEXMF`", values from `./texmf.cnf` override those from `$TEXMF/texmf.cnf`.

While reading the description of the format of the file `texmf.cnf` below, please also refer to p. 95, which lists the `texmf.cnf` file on the CD-ROM.

- Comments start with "`%`" and continue to the end of the line.
- Blank lines are ignored.
- A \ at the end of a line acts as a continuation character, i.e., the next line is appended. Whitespace at the beginning of continuation lines is not ignored.
- Each remaining line must look like

  variable`[.`*progname*`] [=]` *value*

  where the "=" and surrounding whitespace is optional.
- The *variable* name may contain any character other than whitespace, "=", or ".", but sticking to "`A-Za-z_`" is safest.
- If "`.`*progname*" is present, the definition only applies if the program that is running is named *progname* or *progname*`.exe`. This allows different flavors of TEX to have different search paths, for example.
- *value* may contain any characters except "`%`" and "`@`". The "`$`*var*`.`*prog*" feature is not available on the right-hand side; instead, you must use an additional variable (see the definition of the variable `latex2e_inputs` for example). A "`;`" in *value* is translated to ":" if running under Unix; this is useful to write a single `texmf.cnf` which can be used under both Unix and NT.
- All definitions are read before anything is expanded, so you can use variables before they are defined.

A configuration file fragment illustrating most of these points is shown below:

```
% TeX input files -- i.e.,
% anything found by \input or \openin ...
latex209_inputs = \
  .:$TEXMF/tex/latex209//:$TEXMF/tex//
latex2e_inputs  = \
  .:$TEXMF/tex/latex//:$TEXMF/tex//
TEXINPUTS = .:$TEXMF/tex//
TEXINPUTS.latex209 = $latex209_inputs
TEXINPUTS.latex2e = $latex2e_inputs
TEXINPUTS.latex = $latex2e_inputs
```

4.1.3 Path expansion

Kpathsea recognizes certain special characters and constructions in search paths, similar to that in Unix shells. As an general example, the following complex path: `~$USER/{foo,bar}//baz` expands to all subdirectories under directories `foo` and `bar` in `$USER`'s home directory that contain a directory or file `baz`. These expansions are explained in the sections below.

4.1.4 Default expansion

If the highest-priority search path (see "Path sources" on p. 88) contains an *extra colon* (i.e., leading, trailing, or doubled), Kpathsea inserts at that point the next-highest-priority search path that is defined. If that inserted path has an extra colon, the same happens with the next-highest. For example, given an environment variable setting

```
setenv TEXINPUTS /home/karl:
```

and a `TEXINPUTS` value from `texmf.cnf` of

```
.:$TEXMF//tex
```

then the final value used for searching will be:

```
/home/karl:.:$TEXMF//tex
```

Since it would be useless to insert the default value in more than one place, Kpathsea changes only one extra ":" and leaves any others in place: it checks first for a leading ":", then a trailing ":", then a doubled ":".

4.1.5 Brace expansion

A useful feature is brace expansion, which means that, for instance, `v{a,b}w` expands to `vaw:vbw`. Nesting is allowed. This can be used to implement multiple TEX hierarchies, by assigning a brace list to `$TEXMF`.

For example, in `texmf.cnf`, you find the following definition:

```
texdir = $TEXMFLOCAL/tex,!!$TEXMFMAIN/tex
```

Then you can write something like:

```
TEXINPUTS = .;$texdir//
```

which means that after looking in the current directory, first the full `$TEXMFLOCAL/tex` directory tree (on disk) and then the `!!$TEXMFMAIN/tex` tree (using the data base file `ls-R` *only*) will be searched. It is a convenient way for running two parallel TEX structures, one "frozen" (like on a CD-ROM) and the other being continuously updated with new versions as they become available. By using the `$texdir` variable in all definitions, one is sure to always search the up-to-date tree first.

4.1.6 Subdirectory expansion

Two or more consecutive slashes in a path element following a directory *d* is replaced by all subdirectories of *d*: first those subdirectories directly under *d*, then the subsubdirectories under those, and so on. At each level, the order in which the directories are searched is *unspecified*.

If you specify any filename components after the "`//`", only subdirectories with matching components are included. For example, "`/a//b`" expands into directories `/a/1/b`, `/a/2/b`, `/a/1/1/b`, and so on, but not `/a/b/c` or `/a/1`.

Multiple "`//`" constructs in a path are possible, but "`//`" at the beginning of a path is ignored.

4.1.7 List of special characters and their meaning: a summary

The following list summarises the meaning of special characters in Kpathsea configuration files.

`:` Separator in path specification; at the beginning or the end of a path it substitutes the "default" path expansion.

`;` Separator on non-Unix systems (acts like `:`).

`$` Variable expansion.

`~` Represents the user's home directory.

`{...}` Brace expansion, e.g., `a{1,2}b` will become `a1b:a2b`.

`//` Subdirectory expansion. It can occur in the middle or at the end of a path (not at the beginning).

`%` Start of comment.

`\` Continuation character (allows multi-line entries).

`!!` Search *only* database to locate file, *do not* search the disk.

4.2 Filename databases

Kpathsea goes to some lengths to minimize disk accesses for searches. Nevertheless, at installations with enough directories, searching each possible directory for a given file can take an excessively long time (this is especially true if many hundreds of font directories have to be traversed.) Therefore, Kpathsea can use an externally-built "database" file named `ls-R` that maps files to directories, thus avoiding the need to exhaustively search the disk.

A second database file `aliases` allows you to give additional names to the files listed in `ls-R`. This can be helpful to adapt to "8.3" filename conventions in source files.

4.2.1 `ls-R` filename database

As explained above, the name of the main filename database must be `ls-R`. You can put one at the root of each TEX installation hierarchy you wish to search (`$TEXMF` by default); most sites have only one hierarchy. Kpathsea looks for `ls-R` files along the `TEXMFDBS` path.

The recommended way to create and maintain "`ls-R`" is to run the `MakeTeXls-R` script coming with the distribution. It is invoked by the various "`MakeTeX...`" scripts. In principle, this script just runs the command

```
cd /your/texmf/root && ls -LAR ./ >ls-R
```

presuming your system's `ls` produces the right output format (GNU's `ls` is all right). To ensure that the database is always up to date, it is easiest to rebuild it regularly via `cron`, so that for changes in the installed files — perhaps after installing or updating a LATEX package — the file `ls-R` is automatically updated.

If a file is not found in the database, by default Kpathsea goes ahead and searches the disk. If a particular path element begins with "`!!`", however, *only* the database will be searched for that element, never the disk.

4.2.2 `kpsewhich`: Standalone path searching

The `kpsewhich` program exercises path searching independent of any particular application. This can be useful as a sort of `find` program to locate files in TEX hierarchies (this is used heavily in the distributed "`MakeTeX...`" scripts).

```
kpsewhich option... filename...
```

Options can start with either "-" or "--", and any unambiguous abbreviation is accepted.

Kpathsea looks up each non-option argument on the command line as a filename, and returns the first file found. There is no option to return all the files with a particular name (you can run the Unix "`find`" utility for that).

The more important options are described next.

`-dpi=`*num* Set the resolution to *num*; this only affects "`gf`" and "`pk`" lookups. "`-D`" is a synonym, for compatibility with `dvips`. Default is 600.

`-format=`*name*

Set the format for lookup to *name*. By default, the format is guessed from the filename. In fact, the recognized filename extensions and the allowable *name*s (including any leading ".") are the same.

You can also specify an integer for *name*; this is the only way to specify formats that don't have an associated suffix, such as MetaPost support files and `dvips` configuration files. It's also somewhat faster, since no unused formats need to be initialized. The integers appear in the output of "`-help`". Currently recognized file type numbers, with their description, possible file extensions, and the corresponding environment variables (between parentheses[1]) as follows:

```
 0 Generic font files
   .gf (GFFONTS, GLYPHFONTS,
        TEXFONTS)
 1 packed font files
   .pk (PKFONTS, TEXPKS,
        GLYPHFONTS, TEXFONTS)
 2 TeX bitmap font
       (GLYPHFONTS)
 3 Adobe PostScript font metrics
   .afm (AFMFONTS)
 4 METAFONT memory dump
   .base (MFBASES, TEXMFINI)
 5 BibTeX bibliography database
   .bib (BIBINPUTS, TEXBIB)
 6 BibTeX styles
   .bst (BSTINPUTS)
 7 Runtime configuration files
   .cnf (TEXMFCNF)
 8 Web2c filename database
   ls-R (TEXMFDBS)
 9 TeX memory dump
   .fmt (TEXFORMATS, TEXMFINI)
10 TeX generic font maps
   .map (TEXFONTMAPS)
11 MetaPost memory dump
   .mem (MPMEMS, TEXMFINI)
12 METAFONT source files
   .mf  (MFINPUTS)
13 METAFONT program strings
   .pool (MFPOOL, TEXMFINI)
14 METAFONT prettyprinter style files
   .mft (MFTINPUTS)
15 MetaPost sources
   .mp  (MPINPUTS)
16 MetaPost program strings
   .pool (MPPOOL, TEXMFINI)
17 MetaPost support files
        (MPSUPPORT)
18 Ω compiled process
   .ocp (OCPINPUTS)
19 Ω font metrics
   .ofm (OFMFONTS, TEXFONTS)
20 Ω property list
   .opl (OPLFONTS, TEXFONTS)
21 Ω tranlation process files
   .otp (OTPINPUTS)
22 Ω virtual fonts
   .ovf (OVFFONTS, TEXFONTS)
23 Ω virtual property lists
   .ovp (OVPFONTS, TEXFONTS)
24 graphics/figure
   .eps .epsi (TEXPICTS, TEXINPUTS)
25 Source input files read by TeX
   .tex .ltx .dtx .texi .texinfo
   .txi .cls .sty .eps .epsi
        (TEXINPUTS)
26 TeX documentation
   .ps .pdf .doc .txt (TEXDOCS)
27 TeX program strings
   .pool (TEXPOOL, TEXMFINI)
28 TeX system package sources
   .dtx .ins (TEXSOURCES)
29 PostScript header/font
   .pro (TEXPSHEADERS, PSHEADERS)
30 Troff fonts
        (TRFONTS)
31 TeX font metric files
   .tfm (TFMFONTS, TEXFONTS)
32 PostScript type1 fonts
   .pfa .pfb (T1FONTS, T1INPUTS,
              TEXPSHEADERS, PSHEADERS)
33 virtual fonts
   .vf  (VFFONTS, TEXFONTS)
34 dvips configuration files
   config.xxx, xxx.map (TEXCONFIG)
35 MakeIndex style files
  .ist  (TEXIDXSTYLE, INDEXSTYLE)
```

These environment variables are set by default in the configuration file `texmf.cnf`. It is only when you want to override one or more of the values specified in that file that you might want to set them explicitly in your execution environment.

Note that the "`-format`" and "`-path`" options are mutually exclusive.

[1] You can find definitions for these environment variables in the file `texmf.cnf` (p. 95)

`-mode=`*`string`*
: Set the mode name to *string*; this also only affects "`gf`" and "`pk`" lookups. No default: any mode will be found.

`-must-exist`
: Do everything possible to find the files, notably including searching the disk. By default, only the `ls-R` database is checked, in the interest of efficiency.

`-path=`*`string`*
: Search along the path *string* (colon-separated as usual), instead of guessing the search path from the filename. "`//`" and all the usual expansions are supported. The options "`-path`" and "`-format`" are mutually exclusive.

`-progname=`*`name`*
: Set the program name to *name*. This can affect the search paths via the "*`.progname`*" feature in configuration files. The default is "`kpsewhich`".

`-show-path=`*`name`*
: shows the path used for file lookups of file type *name*. Either a filename extension ("`.pk`", "`.vf`", etc.) or an integer can be used, just as with "`-format`" option.

`-debug=`*`num`*
: sets the debugging options to *num*.

4.2.3 Examples of use

Let us now have a look at Kpathsea in action.

```
>> kpsewhich -format=.tex article.cls
/usr/local/share/texmf/tex/latex/base/article.cls
```

We are looking for the file `article.cls` in the TEX source file directories (type `.tex`, format type 25). We find it in the subdirectory `tex/latex/base` below the "`TEXMF`" root directory. To save space, in the following examples we will denote with ... the repetitive part `/usr/local/share/texmf` preceding each file path.

```
>> kpsewhich tugboat.bib
.../bibtex/bib/beebe/tugboat.bib
```

BIBTEX bibliography databases correspond to format type `.bib`. Here we located file `tugboat.bib`.

```
>> kpsewhich cmr10.pk
.../fonts/pk/ljfour/public/cm/cmr10.600pk

>> kpsewhich -dpi=300 cmr10.pk

>> kpsewhich ptmb8r.pk
.../fonts/pk/modeless/dpi597/ptmb8r.pk

>> kpsewhich -dpi=300 ptmb8r.pk
.../fonts/pk/modeless/dpi300/ptmb8r.pk
```

Font bitmap glyph files of type `.pk` correspond to format type 2. They are used by visualization programs like `dvips` and `xdvi`. On our system we found the Computer Modern file `cmr10` for the mode `ljfour`, at a base resolution of 600 dpi (dots per inch). However, when specifying that we are only interested in a resolution of 300dpi (`-dpi=300`) we are told there is no such font available on the system. In fact, a program like `dvips` or `xdvi` would go off and actually build the `.pk` files at the required resolution using the script `MakeTeXPK`. The last two commands look for a file `ptmb8r.pk`. When specifying no explicit resolution the system returns one (at 597 dpi) which is closest to the "default" set in the `MakeTeXPK` script (600 dpi). However, when specifying the desired resolution (300 dpi) the full path name of the relevant target file is shown.

Next we turn our attention to `dvips`'s header (format type 29) and configuration files (format type 34).

```
>> kpsewhich tex.pro
.../dvips/base/tex.pro

>> kpsewhich -format=34  psfonts.map
.../dvips/base/psfonts.map

>> kpsewhich -format=.map config.ps
.../dvips/config/config.ps
```

We first look at a few of the commonly used files, namely the general prolog `tex.pro` for TEX support, before turning our attention to the generic configuration file (`config.ps`) and the PostScript font map `psfonts.map`. Note how we fool the system by asking for `config.ps` as if it had a suffix of `.map`.

We now look a little closer at the URW Times PostScript support files. The name for these in Berry's font naming scheme is "`utm`". The first file we look at is the configuration file, which contains the name of the map file.

```
>> kpsewhich -format=34  config.utm
.../dvips/config/config.utm
```

The contents of that file is

```
p +utm.map
```

which points to the file `utm.map`, which we want to locate next.

```
>> kpsewhich utm.map
.../dvips/urw/utm.map
```

In this map file, which resides in `dvips`'s `urw` subdirectory, the file names of the Type1 PostScript

fonts referenced are defined. The contents looks like (we only show part of the lines):

```
utmb8r NimbusRomNo9L-Medi      ... <utmb8a.pfb
utmbi8r NimbusRomNo9L-MediItal... <utmbi8a.pfb
utmr8r NimbusRomNo9L-Regu      ... <utmr8a.pfb
utmri8r NimbusRomNo9L-ReguItal... <utmri8a.pfb
utmbo8r NimbusRomNo9L-Medi "   ... <utmb8a.pfb
utmro8r NimbusRomNo9L-Regu "   ... <utmr8a.pfb
```

Let's, for instance take the Times Regular instance `utmr8a.pfb`, and find its position in the `texmf` directory tree by using a search with format type 32.

```
>> kpsewhich utmr8a.pfb
.../fonts/type1/urw/utm/utmr8a.pfb
```

It should be evident from these few examples how one can easily locate the whereabouts of a given file. This is especially important if you suspect that the wrong version of a file is picked up somehow, since `kpsewhich` will show you the first file encountered.

4.2.4 Debugging actions

Sometimes it is necessary to really investigate how a program resolves file references. To make this feasible in a convenient way Kpathsea offers various debug levels:

- 1 `stat` calls (file tests). When running with an up-to-date `ls-R` database this should almost give no output.
- 2 References to hash tables (like `ls-R` database, map files, configuration files).
- 4 File open and close operations.
- 8 General path information for file types searched by Kpathsea. This is useful to find out where a particular path for the file was defined.
- 16 Directory list for each path element (only relevant for searches on disk).
- 32 File searches.

A value of `-1` will set all the above options; in practice you will probably always use these levels if you need any debugging.

Similarly, with the `dvips` program one can, by setting some debug switches, follow in detail where files are picked up from. Alternatively, when a file is not found, the debug trace shows in which directories the program looks for the given file, so that one can get an indication what the problem is.

Generally speaking, as most programs call the Kpathsea library internally, you can select a debug option by using the `KPATHSEA_DEBUG` environment variable, and setting it to (a combination of) values as described in the above list.

Let us consider, as an example, a small LaTeX source file, `hello_world.tex`, which contains the following input.

```
\documentclass{article}
\begin{document}
Hello World!
\end{document}
```

This little file only used the font `cmr10`, so let us look how `dvips` prepares the PostScript file.

```
>> dvips -d4100 hello_world -o
```

In this case we have combined `dvips`'s debug class 4 (font paths) with Kpathsea's path element expansion (see `dvips` Reference Manual). We get something like shown below (we have rearranged the output for easier display).

```
debug:start search(file=texmf.cnf, must_exist=1,
                   find_all=1,
  path=.:/usr/local/bin/texlive:/usr/local/bin:
       /usr/local/bin/share/texmf/web2c:/usr/local:
       /usr/local/share/texmf/web2c:
       /.:/./teTeX/TeX/share/texmf/web2c:).
kdebug:start search(file=ls-R, must_exist=1,
                   find_all=1,
  path=/usr/local/texmf:/usr/local/share/texmf).
kdebug:search(ls-R) =>/usr/local/share/texmf/ls-R
kdebug:start search(file=aliases, must_exist=1,
                   find_all=1,
  path=/usr/local/texmf:/usr/local/share/texmf).
kdebug:search(aliases) =>
kdebug:start search(file=config.ps, must_exist=0,
                   find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//).
kdebug:search(config.ps) =>
  /usr/local/share/texmf/dvips/config/config.ps
kdebug:start search(file=/root/.dvipsrc,
                   must_exist=0, find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//).
kdebug:search($HOME/.dvipsrc) =>
   ...
kdebug:start search(file=psfonts.map, must_exist=0,
                   find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//).
kdebug:search(psfonts.map) =>
  /usr/local/share/texmf/dvips/base/psfonts.map
```

First `dvips` locates its working files. It first found `texmf.cnf` (with the definitions of the paths of the other files), then the file data base `ls-R` (to optimize file searching). It goes on to find the generic configuration file `config.ps`, and then looks for the customization file `.dvipsrc` (which, in this case is *not found*). Finally `dvips` locates the generic map file for PostScript fonts `psfonts.map` (defining the relation between the internal and external names for the PostScript fonts).

At this point `dvips` identifies itself to the user:

```
dvipsk 5.66a Copyright 1986-97 Radical Eye Software
                               (www.radicaleye.com)
```

then goes on to look for the prolog file `texc.pro`,

```
kdebug:start search(file=texc.pro, must_exist=0,
                    find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//:
       /usr/local/texmf/fonts//type1//:
       !!/usr/local/share/texmf/fonts//type1//).
kdebug:search(texc.pro) =>
  /usr/local/share/texmf/dvips/base/texc.pro
```

After having found the file, `dvips` outputs date and time, and informs us that it will generate the file `hello_world.ps`, then that it needs the font file `cmr10`, and that the latter is declared as "resident"

```
' TeX output 1997.05.01:1316' -> hello_world.ps
Defining font () cmr10 at 10.0pt
Font cmr10 <CMR10> is resident.
```

Now the search is on for the file `cmr10.tfm`, which is found, then a few more prolog files (not shown), and finally for the Type1 instance `cmr10.pfb` of the font (which is found) and included in the output file (see last line).

```
kdebug:start search(file=cmr10.tfm, must_exist=1,
                    find_all=0,
  path=.:/usr/local/texmf/fonts/tfm//:
       !!/usr/local/share/texmf/fonts/tfm//:
       /var/tex/fonts/tfm//).
kdebug:search(cmr10.tfm) =>
/usr/local/share/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must_exist=0,
                    find_all=0,
   ...
<texps.pro>.

kdebug:start search(file=cmr10.pfb, must_exist=0,
                    find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//:
       /usr/local/texmf/fonts//type1//:
       !!/usr/local/share/texmf/fonts//type1//).
kdebug:search(cmr10.pfb) =>
/usr/local/share/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]
```

4.3 Runtime options

Another of the nice features of Web2c 7.0 is its possibility to control a number of memory parameters (in particular, array sizes) via the runtime file `texmf.cnf` read by Kpathsea. A detailed list of all set-table parameters can be found in that file (see p. 95, Part 3 starting at line 261). The most interesting values are:

`main_memory` Total words of memory available, for TEX, METAFONT, and MetaPost. You must make a new format file for each different setting. For instance, you could generate a "huge" version of TEX, and call the format file `hugetex.fmt`. Using the standard way of specifying the program name used by Kpathsea the particular value of the `main_memory` variable will then be read from `texmf.cnf` (See p. 95, line 280 for the generic value and line 281 for the "huge" one instantiated by `hugetex`).

`extra_mem_bot` Extra space for "large" TEX data structures: boxes, glue, breakpoints, etc. Especially useful if you use PICTEX.

`font_mem_size` Number of words for font information available for TEX. This is more or less the total size of all TFM files read.

`hash_extra` Additional space for the hash table of control sequence names. Approximately 10,000 control sequences can be stored in the main hash table; if you have a large book with numerous cross-references, this might not be enough. On line 297 and 298 of file `texmf.cnf` as shown in p. 95 you see that both the `hugetex` and `pdftex` program invocations ask for an extra 10,000 control sequences (the default value of `hash_extra` is zero, as seen on line 296).

Of course, this facility is no substitute for truly dynamic arrays and memory allocation, but since this is extremely difficult to implement in present TEX, these runtime parameters provide a practical compromise allowing some flexibility.

5 Other packages on the CD-ROM

While the main portion of **TEX Live** (the fonts, macros and documentation) can be used on any TEX system, the set of runnable binaries is not suitable for everyone. To make the disk as widely useful as possible, we have included the original distributions of four complete TEX systems, two for Macintosh, one for Windows 95, and one for DOS and OS/2. Windows 3.1 users should look at the GUTenberg distribution on the CD-ROM.

5.1 OzTEX[2]

OzTEX is a Macintosh TEX system created by Andrew Trevorrow. The OzTEX application includes TEX, `INITEX`, a DVI previewer, a DVI-to-PostScript translator (Tom Rokicki's `dvips`) and a driver for QuickDraw printers. OzTEX also includes `dvidvi`, `dvicopy`, and Angus Duggan's PostScript utilities: `psbook`, `psnup`, `psselect` and `pstops`.

The version of `dvips` included in OzTEX supports HyperTEX and the partial downloading of PostScript fonts. It has also been enhanced for Mac users in a number of ways: Standard Mac PostScript fonts (LWFN files) can be downloaded, fully or partially. All OzTEX-specific `\special` commands are supported, such as the inclusion of

[2] This section was written by Andrew Trevorrow.

PICT/PNTG/EPSF files. The `dvips` output can be sent directly to the current printer.

OzTEX's previewer has lots of features to make it easy to proofread DVI files. It can handle PK and PostScript fonts. Anti-aliasing is supported. Virtual fonts are processed on the fly. The previewer supports most of the `\special` commands generated by LATEX's `color`, `graphics/x` and `hyperref` packages. It recognizes all `dvips`-specific `\special`s and those it cannot handle (like rotation) are silently ignored.

OzTEX includes all the most popular formats and macro packages. Plain TEX, LATEX, AMS-TEX, AMS-LATEX and REVTEX are all installed and ready to run.

OzTEX is easy to extend and customize. A default configuration file is read when OzTEX starts up; it contains a host of parameters for setting up search paths, telling TEX how much memory to allocate for various arrays, specifying which TFMs are for PostScript fonts, etc. A Config menu makes it easy to load other config files at any time. And for even more flexibility, OzTEX can automatically load a specified config file just before typesetting, previewing or printing.

5.1.1 Additional programs

The usual assortment of TEX-related programs are provided with OzTEX, including OzMF, a Mac implementation of METAFONT, and OzMP, a Mac port of John Hobby's MetaPost program for producing PostScript pictures using a METAFONT-like language.

The following programs are also distributed with OzTEX, courtesy of their authors; BibTEX by Vince Darley; MakeIndex by Rick Zaccone; Excalibur, a TEX/LATEX spelling checker, by Rick Zaccone and Robert Gottshall; and AlphaLite, a TEX/LATEX-savvy text editor, by Pete Keleher.

For the latest information about OzTEX, keep an eye on the Web page at the URL `http://www.kagi.com/authors/akt/oztex.html`.

An even better way to keep up-to-date is to join the `oztex-info` mailing list. To subscribe, send some e-mail to

```
majordomo@maths.adelaide.edu.au
```

with the following line in the *body* of the message:

```
subscribe oztex-info
```

OzTEX is distributed as shareware, so you are welcome to try it out before paying the registration fee. The individual fee is US$30 and the site fee is US$300. See the "Shareware Fee" item in OzTEX's Help menu for details on how to pay. E-mail support is provided to registered users. Send all queries and comments to Andrew Trevorrow (`akt@kagi.com`).

5.2 CMacTeX[3]

CMacTeX is an implementation of TEX for the Macintosh by Thomas Kiffe (`mailto:tkiffe@math.tamu.edu`). It includes the three main parts of any TEX installation — TEX, METAFONT and dvips. It also includes two dvi previewers, a utility for printing dvi files on a non PostScript printer, a PostScript previewer and numerous utilities for manipulating TEX fonts. Full support for the automatic generation of pk font files is an integral part of the distribution. CMacTeX can be configured to work in an integrated fashion with BBEdit, Alpha, and MPW. It will run on any Macintosh with 8 MB of RAM and System 7.

CMacTeX is shareware. The registration fee is US$35 for a single-user license and US$150 for a site license.

Installation instructions can be found in the file `/macintosh/cmactex/ReadMeFirst`

5.3 MiKTeX[4]

MiKTeX 1.07 is an implementation by Christian Schenk (`mailto:cschenk@berlin.snafu.de`) of TEX and METAFONT related utilities for Windows NT and Windows 95. The MiKTeX distribution includes TEX; LATEX 2ε Dec'96 including standard packages; METAFONT; MetaPost; dvips MakeIndex; BIBTEX; YAP (Yet Another Previewer); TeXware (dvitype etc.); METAFONTware (gftopk etc.); psutils (psselect, pstops etc.); and DVIcopy.

Installation instructions can be found in the file `/win32/miktex/README.TXT`

5.4 emTeX

The emTeX distribution for DOS and OS/2 is wriiten by Eberhard Mattes (`mailto:mattes@azu.informatik.uni-stuttgart.de`). It includes the TEX typesetter, the METAFONT font generation program, printer drivers, screen previewers, and tools like BIBTEX and MakeIndex. It also includes the macro packages LATEX 2.09 and LATEX 2ε. Fonts are included as pixel files and METAFONT source files.

Installation instructions can be found in the file `/msdos/emtex/README.ENG`

[3] This section is taken from the CMacTeX documentation.

[4] This section is drawn from the documentation.

A The texmf.cnf file

```
% original texmf.cnf -- runtime path configuration file for kpathsea.
% (If you change or delete 'original' on the previous line, the
% distribution won't install its version over yours.)
% Public domain.
%
% What follows is a super-summary of what this .cnf file can
% contain. Please read the Kpathsea manual for more information.
%
% texmf.cnf is generated from texmf.cnf.in, by replacing @var@ with the
% value of the Make variable 'var', via a sed file texmf.sed, generated
% (once) by kpathsea/Makefile (itself generated from kpathsea/Makefile.in
% by configure).
%
% Any identifier (sticking to A-Za-z_ for names is safest) can be assigned.
% The '=' (and surrounding spaces) is optional.
% No % or @ in texmf.cnf.in, for the sake of autogeneration.
% (However, %'s and @'s can be edited into texmf.cnf or put in envvar values.)
% $foo (or ${foo}) in a value expands to the envvar or cnf value of foo.
%
% Earlier entries (in the same or another file) override later ones, and
% an environment variable foo overrides any texmf.cnf definition of foo.
%
% All definitions are read before anything is expanded, so you can use
% variables before they are defined.
%
% If a variable assignment is qualified with '.PROGRAM', it is ignored
% unless the current executable (last filename component of argv[0]) is
% named PROGRAM.  This foo.PROGRAM construct is not recognized on the
% right-hand side. For environment variables, use FOO_PROGRAM.
%
% Which file formats use which paths for searches is described in the
% various programs' and the kpathsea documentation.
%
% // means to search subdirectories (recursively).
% A leading !! means to look only in the ls-R db, never on the disk.
% A leading/trailing/doubled : in the paths will be expanded into the
%   compile-time default. Probably not what you want.

%  Part 1: Search paths and directories.

% The root of everything below.
prefix = $SELFAUTOPARENT

% You can set an environment variable to override this if you're testing
% a new TeX tree, without changing anything else.
%
% You may wish to use one of the $SELFAUTO... variables here so TeX will
% find where to look dynamically.  See the manual and the definition
% below of TEXMFCNF.
%
% If you have multiple trees, you can use shell brace notation, for example:
% TEXMF = {/usr/local/mytex,/usr/local/othertex}
% and also set TEXMFDBS to /usr/local/mytex:/usr/local/othertex.
TEXMFLOCAL=/usr/local/texmf
TEXMFMAIN = $prefix/share/texmf
TEXMFLS_R = $TEXMFLOCAL
% Where to look for ls-R files.  There need not be an ls-R in the
```

```
% directories in this path, but if there is one, Kpathsea will use it.
TEXMFDBS = $TEXMFLOCAL;$TEXMFMAIN

% Where you want generated files to go. Choose one of the texmf trees
% listed in $TEXMF. The following is the default:
VARTEXMF = $TEXMFLOCAL

% The TeX inputs and fonts directories.
texdir = {$TEXMFLOCAL/tex,!!$TEXMFMAIN/tex}
omegadir ={$TEXMFLOCAL/omega,!!$TEXMFMAIN/omega}
fontdir = {$TEXMFLOCAL/fonts,!!$TEXMFMAIN/fonts}
dbmain=!!$TEXMFMAIN
%
dbtex = $texdir
dbomega = $omegadir
dbfonts = $fontdir

% TeX input files -- i.e., anything to be found by \input or \openin,
% including .sty, .eps, etc.
TEXINPUTS = .;$dbtex//

% LaTeX 2e specific macros are stored in latex.
% latex209 is not supported, at the request of the authors of LaTeX
%
latex_inputs = .;$dbtex/latex//;$dbtex/generic//
TEXINPUTS.latex = $latex_inputs
TEXINPUTS.latextex = $latex_inputs

% Omega
TEXINPUTS.lambda2e = .;$dbomega/lambda//;$latex_inputs
TEXINPUTS.lambda = .;$dbomega/lambda//;$latex_inputs

% Fontinst needs to read afm files.
TEXINPUTS.fontinst = .;$dbtex//;$dbfonts/afm//

% Plain TeX.  Have the command tex check all directories as a last
% resort, we may have plain-compatible stuff anywhere.
plain_inputs = $dbtex/plain//;$dbtex/generic//
TEXINPUTS.textex = .;$dbtex//;$plain_inputs
TEXINPUTS.tex = .;$plain_inputs;$dbtex//
TEXINPUTS.omega = .;$dbomega//;$plain_inputs;$dbtex//

% INITEX. May as well make this separate so it can search on disk;
% initex is seldom run, and might be used directly after files have been
% added, when ls-R may not up be to date.
TEXINPUTS.initex = .;$texdir//
TEXINPUTS.iniomega = .;$omegadir//;$texdir//

% Earlier entries override later ones, so put this last.
TEXINPUTS.viromega = .;$dbomega//;$dbtex//
TEXINPUTS = .;$dbtex//

% Metafont, MetaPost inputs.
MFINPUTS = .;{$TEXMFLOCAL,$dbmain}/metafont//;{$dbfonts,$VARTEXFONTS}/source//
MPINPUTS = .;{$TEXMFLOCAL,$dbmain}/metapost//

% Dump files (fmt/base/mem) for vir{tex,mf,mp} to read (see
% web2c/INSTALL), and string pools (.pool) for ini{tex,mf,mp}.  It is
% silly that we have six paths and directories here (they all resolve to
```

```
% a single place by default), but historically ...
TEXFORMATS = .;{$TEXMFLOCAL,$dbmain}/web2c
MFBASES = .;{$TEXMFLOCAL,$dbmain}/web2c
MPMEMS = .;{$TEXMFLOCAL,$dbmain}/web2c
TEXPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c
MFPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c
MPPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c

% If you have a read-only central font directory and therefore need to
% cache MakeTeXPK-created fonts locally, set this as appropriate and
% also set the 'vartexfonts' feature in MT_FEATURES in MakeTeXnames.cnf.
VARTEXFONTS = /var/tex/fonts

% Device-independent font metric files.
VFFONTS = .;$dbfonts/vf//
TFMFONTS = .;$dbfonts/tfm//;$VARTEXFONTS/tfm//

% The $MAKETEX_MODE below means the drivers will not use a cx font when
% the mode is ricoh.  If no mode is explicitly specified, kpse_prog_init
% sets MAKETEX_MODE to /, so all subdirectories are searched.  See the manual.
PKFONTS = .;{$dbfonts,$VARTEXFONTS}/pk/$MAKETEX_MODE//

% xdvi needs to find bitmaps for PostScript fonts, which can be
% generated by the standalone program gsftopk, among others.
PKFONTS.xdvi = .;{$dbfonts,$VARTEXFONTS}/pk/{$MAKETEX_MODE,modeless}//

% Similarly for the GF format, which only remains in existence because
% Metafont outputs it (and MF isn't going to change).
GFFONTS = .;$dbfonts/gf/$MAKETEX_MODE//

% A backup for PKFONTS and GFFONTS. Not used for anything.
GLYPHFONTS = .;$dbfonts

% For texfonts.map and included map files used by MakeTeXPK.
% See ftp;//ftp.tug.org/tex/fontname.tar.gz.
TEXFONTMAPS = .;{$TEXMFLOCAL,$dbmain}/fontname

% BibTeX bibliographies and style files.
BIBINPUTS = .;{$TEXMFLOCAL,$dbmain}/bibtex/bib//
BSTINPUTS = .;{$TEXMFLOCAL,$dbmain}/bibtex/bst//

% MFT style files.
MFTINPUTS = .;{$TEXMFLOCAL,$dbmain}/mft//

% PostScript headers, prologues (.pro), encodings (.enc) and fonts.
TEXPSHEADERS = .;$TEXMFLOCAL/dvips;$dbmain/dvips//;$dbfonts/type1//

% PostScript Type 1 outline fonts.
T1FONTS = .;$dbfonts/type1//;$TEXMFLOCAL/dvips//;$dbmain/dvips//

% PostScript AFM metric files.
AFMFONTS = .;$dbfonts/afm//

% Dvips' config.* files (this name should not start with 'TEX'!).
TEXCONFIG = .;$TEXMFLOCAL/dvips;$dbmain/dvips//

% Makeindex style (.ist) files.
INDEXSTYLE = .;{$TEXMFLOCAL,$dbmain}/makeindex//

```

```
% Used by DMP (ditroff-to-mpx), called by makempx -troff.
TRFONTS = /usr/lib/font/devpost
MPSUPPORT = .;{$TEXMFLOCAL,$dbmain}/metapost/support

% For xdvi to find mime.types and .mailcap, if they do not exist in
% $HOME.  These are single directories, not paths.
% (But the default mime.types, at least, may well suffice.)
MIMELIBDIR = $prefix/etc
MAILCAPLIBDIR = $prefix/etc

% TeX documentation and source files, for use with kpsewhich.
TEXDOCS = .;{$TEXMFLOCAL,$dbmain}/doc//
TEXSOURCES = .;{$TEXMFLOCAL,$dbmain}/source//

% Omega-related fonts and other files.
OFMFONTS = .;{$dbfonts,$VARTEXFONTS}//ofm//
OPLFONTS = .;{$dbfonts,$VARTEXFONTS}//opl//
OVFFONTS = .;{$dbfonts,$VARTEXFONTS}//ovf//
OVPFONTS = .;{$dbfonts,$VARTEXFONTS}//ovp//
OTPINPUTS = .;{$TEXMFLOCAL,$dbmain}/omega/otp//
OCPINPUTS = .;{$TEXMFLOCAL,$dbmain}/omega/ocp//

% For MakeTeX.common, MakeTeX.site, ls-R.
web2cdir = {$TEXMFLOCAL,$dbmain}/web2c
TEXMFCNF_DIR = $TEXMFMAIN/web2c

% The MakeTeX* scripts rely on KPSE_DOT. Do not change it.
KPSE_DOT = .

% This definition isn't used from this .cnf file itself (that would be
% paradoxical), but the compile-time default in paths.h is built from it.
% The SELFAUTO* variables are set automatically from the location of
% argv[0], in kpse_set_progname.
%
% About the /. construction;
% 1) if the variable is undefined, we'd otherwise have an empty path
%    element in the compile-time path.  This is not meaningful.
% 2) if we used /$VARIABLE, we'd end up with // if VARIABLE is defined,
%    which would search the entire world.
%
% The TETEXDIR stuff isn't likely to relevant unless you're using teTeX,
% but it doesn't hurt.
%
TEXMFCNF =.;$SELFAUTOLOC;$SELFAUTODIR;$SELFAUTODIR/share/texmf/web2c;$SELFAUTOPARENT;\
$SELFAUTOPARENT/share/texmf/web2c;/.$TETEXDIR;/.{$TEXMFLOCAL,$dbmain}/web2c;$web2cdir

%  Part 2; Non-path options.

% Write .log/.dvi/etc. files here, if the current directory is unwritable.
% TEXMFOUTPUT = /tmp

% If a dynamic file creation fails, log the command to this file, in
% either the current directory or TEXMFOUTPUT.  Set to the
% empty string or  0  to avoid logging.
MISSFONT_LOG = missfont.log

% Set to a colon-separated list of words specifying warnings to suppress.
% To suppress everything, use TEX_HUSH = all; this is equivalent to
% TEX_HUSH = checksum;lostchar;readable;special
```

```
TEX_HUSH = 0

% Enable system commands via \write18{...}?
shell_escape = 0

% Allow TeX \openout on filenames starting with `.' (e.g., .rhosts)?
openout_any = 0

% Enable the MakeTeX... scripts by default?  These must be set to 0 or 1.
% Particular programs can and do override these settings, for example
% dvips's -M option.  Your first chance to specify whether the scripts
% are invoked by default is at configure time.
%
% These values are ignored if the script names are changed; e.g., if you
% set DVIPSMAKEPK to `foo', what counts is the value of the environment
% variable/config value `FOO', not the `MAKETEXPK' value.
%
% MAKETEXTEX = 0
% MAKETEXPK = 0
% MAKETEXMF = 0
% MAKETEXTFM = 0

% What MetaPost runs to make MPX files.  This is passed an option -troff
% if MP is in troff mode.  Set to `0' to disable this feature.
MPXCOMMAND = makempx

%  Part 3; Array and other sizes for TeX (and Metafont and MetaPost).
%
% If you want to change some of these sizes only for a certain TeX
% variant, the usual dot notation works, e.g.,
% main_memory.hugetex = 20000000
%
% If a change here appears to be ignored, try redumping the format file.

% Memory. Must be less than 8,000,000.
%
% main_memory is relevant only to initex, extra_mem_* only to non-ini.
% Thus, have to redump the .fmt file after changing main_memory; to add
% to existing fmt files, increase the other.  (To get an idea of how
% much, try \tracingstats=2 in your TeX source file;
% web2c/tests/memtest.tex might also be interesting.)
%
% To increase space for boxes (as might be needed by, e.g., PiCTeX),
% increase extra_mem_bot.
%
main_memory = 263000 % words of memory available; also applies to mf&mp
main_memory.hugetex = 1100000
extra_mem_top = 0    % extra high memory for chars, tokens, etc.
extra_mem_bot = 0    % extra low memory for boxes, glue, breakpoints, etc.

% Words of font info for TeX (total size of all TFM files, approximately).
font_mem_size = 100000
font_mem_size.hugetex = 400000
font_mem_size.pdftex = 400000

% Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
font_max = 500
font_max.hugetex = 900

```

```
% Extra space for the hash table of control sequences (which allows 10K
% names as distributed).
hash_extra = 0
hash_extra.hugetex = 10000
hash_extra.pdftex = 10000

% Max number of characters in all strings, including all error messages,
% help texts, font names, control sequences.  These values apply to TeX and MP.
pool_size = 125000
pool_size.hugetex = 500000
pool_size.pdftex = 500000
% Minimum pool space after TeX/MP's own strings; must be at least
% 25000 less than pool_size, but doesn't need to be nearly that large.
string_vacancies = 25000
string_vacancies.hugetex = 45000
string_vacancies.pdftex = 45000
max_strings = 15000             % max number of strings
max_strings.hugetex = 55000             % max number of strings
max_strings.pdftex = 55000              % max number of strings
pool_free = 5000                % min pool space left after loading .fmt

% Hyphenation trie. As distributed, the maximum is 65535; this should
% work unless 'unsigned short' is not supported or is smaller than 16
% bits.  This value should suffice for UK English, US English, French,
% and German (for example).  To increase, you must change
% 'ssup_trie_opcode' and 'ssup_trie_size' in tex.ch (and rebuild TeX);
% the trie will then consume four bytes per entry, instead of two.
%
% US English, German, and Portuguese; 30000.
% German; 14000.
% US English; 10000.
%
trie_size = 64000

hyph_size = 1000        % number of hyphenation exceptions, >610 and <32767.
buf_size = 3000         % max length of input lines or control sequence names
nest_size = 100         % simultaneous semantic levels (e.g., groups)
max_in_open = 15        % simultaneous input files and error insertions
param_size = 500        % simultaneous macro parameters
save_size = 4000        % for saving values outside current group
save_size.pdftex = 30000        % for saving values outside current group
save_size.hugetex = 30000       % for saving values outside current group
stack_size = 300        % simultaneous input sources

% These work best if they are the same as the I/O buffer size, but it
% doesn't matter much.  Must be a multiple of 8.
dvi_buf_size = 16384 % TeX
gf_buf_size = 16384  % MF

% It's probably inadvisable to change these. At any rate, we must have:
% 45 < error_line      < 255;
% 30 < half_error_line < error_line - 15;
%      max_print_line  < 60 ;
% These apply to Metafont and MetaPost as well.
error_line = 79
half_error_line = 50
max_print_line = 79
```

B Catalogue of Packages

Table 1: **TEX Live** packages

Package	*Collection*	*Description*
a4	latex3	Originally for LaTeX 2.09 but updated for LaTeX 2ε. Mostly superseded by LaTeX 2ε support for a4 paper but defines the extra option of widemargins. The geometry package is usually better.
aiaa	latex3	A bundle of LaTeX/BibTeX files and sample documents to aid those producing papers and journal articles according to the guidelines of the American Institute of Aeronautics and Astronautics (AIAA)
achemso	latex3	LaTeX and BibTeX style for American Chemical Society
acronym	latex3	This package ensures that all acronyms used in the text are spelled out in full at least once. It also provides an environment to build a list of acronyms.
adrlist	latex3	Using address lists in LaTeX.
aguplus	latex3	Styles for American Geophysical Union.
alatex	formats3	An extended LaTeX with better modularity
alg	latex3	LaTeX environments for typesetting algorithms
algorithms	latex3	Defines a floating algorithm environment designed to work with the algorithmic package.
alpha-linux	systems1	System binaries for Alpha running Linux.
alpha-osf3.2	systems1	System binaries for Alpha running OSF 3.2.
altfont	latex3	A generalised replacement for some parts of psnfss and mfnfss. Similar to psfont with the PostScript specific code removed.
amiweb2c	systems1	An Amiga port of the complete UNIX-TeX system.
amsfonts	ams2	A set of miscellaneous TeX fonts from the American Mathematical Society that augment the standard set normally distributed with TeX. The set includes: Extra mathematical symbols; Blackboard bold letters (uppercase only); Fraktur letters; Subscript sizes of bold math italic and bold Greek letters; Subscript sizes of large symbols such as sum and product; Added sizes of the Computer Modern small caps font; Cyrillic fonts (from the University of Washington); Euler math fonts.
amslatex	ams2	A collection of loosely related files that are distributed together by the American Mathematical Society. These files are miscellaneous enhancements to LaTeX whose aim is superior information structure of mathematical documents and superior printed output.
amstex	ams2	American Mathematical Society plain TeX macros
answers	latex3	Styles for setting questions (or exercises) and answers.
apa	latex3	LaTeX class and BibTeX style used to format text according to the American Psychological Association Publication Manual (4th ed.) specifications for manuscripts or, with an option to the package, in an APA journal style format or as a regular document.
apl	fonts3	Fonts for typesetting APL programs.
arabtex	lang3	Macros and fonts for typesetting Arabic
arseneau	latex2	Miscellaneous macros by Donald Arseneau.
ascii	fonts3	Support for IBM extended ASCII font.
astro	fonts3	Astronomical (planetary) symbols.
aurora	dvips3	Header files for dvips to make colour separations
autotab	latex3	Generating tabular setups.
babel	latex1	Multilingual support for LaTeX.
backgammon	fonts3	Style for typesetting backgammon boards.
bakoma	fonts2	Computer Modern and AMS fonts in PostScript Type1 form.
barcodes	fonts3	Fonts for making barcodes.
barr	graphics3	Diagram macros by Michael Barr.
base	latex1	Basic LaTeX system.
bbding	fonts3	An NFSS-interface to the symbol font bbding containing many of the Zapf dingbats fonts.

Catalogue of Packages *continued*

Package	*Collection*	*Description*
bbm	fonts3	Blackboard variant fonts for Computer Modern, with LaTeX support
bbtbase	bibtex1	Basic BibTeX styles.
bbtdoc	bibtex2	Basic BibTeX documentation/
beebe	bibtex2	Nelson Beebe's collection of TeX-related bibliographies, and BibTeX style files
beton	latex3	Typeset a LaTeX 2_ε document with the Concrete fonts designed by Don Knuth and used in his book "Concrete Mathematics".
biblist	latex3	BibTeX styles by Joachim Schrod.
blue	formats3	Kees van der Laan's BLUe format, a concise but expressive document preparation system modelled on Knuth's manmac
bm	latex2	This is a package for accessing bold symbols in math mode. (Similar to the AMS `\boldsymbol` command, but taking more care over spacing, delimiters etc.).
booktabs	latex2	Nicer layout of tables
borceux	graphics3	Diagram macros by Francois Borceux
bridge	latex3	Macros for typesetting Bridge diagrams.
calc	latex2	Adds infix expressions to perform arithmetic in the LaTeX commands `\setcounter`, `\addtocounter`, `\setlength`, and `\addtolength`.
calendar	plain3	Plain macros for making nice calendars
calrsfs	latex3	Nicer calligraphic letters.
camel	latex3	Comprehensive bibliography manager (prototype citation engine for LaTeX3). Will become BibTeX 1.0 on release. Under development.
caption	latex2	Extends caption capabilities for figures and tables, such as the caption width, style, font. Many aspects are tunable as options.
carlisle	latex2	Miscellaneous small packages by David Carlisle
ccfonts	latex	LaTeX support for Concrete fonts.
cchess	fonts3	Macros and fonts for typesetting Chinese Chess board diagrams.
cellular	plain3	Cellular table construction
changebar	latex2	Generate changebars in LaTeX documents.
chemcono	latex3	A LaTeX style file for using compound numbers in chemistry documents. It works like `\cite` and the `\thebibliography`, using `\fcite` and `\theffbibliography` instead. It allows compound names in documents to be numbered and does not affect the normal citation routines.
chemsym	latex3	Macros for typing chemical symbols
cheq	fonts3	Adobe chess font.
cherokee	fonts3	Fonts for Cherokee scripts
chess	fonts3	Fonts for typesetting chess boards.
circ	graphics3	Macros for typesetting circuit diagrams. Several electrical symbols like resistor, capacitor, transistors etc., are defined. The symbols can be connected with wires.
circuit	graphics3	Language for drawing circuit diagrams.
cirth	fonts3	Fonts for Cirth
cite	latex2	Supports compressed, sorted lists of numerical citations: [8,11–16].
cm	fonts1	Computer Modern fonts
cmbright	fonts2	Computer Modern Bright fonts
cmcyralt	latex3	Alternative Russian encoding support
cmextra	fonts2	Extra Computer Modern fonts, from the American Mathematical Society
cmpica	fonts3	A Computer Modern Pica variant
cmps	fonts2	Type1 versions of PostScript fonts, from Blue Sky and Y&Y.
codepage	latex3	Support for variant code pages.
colorsep	dvips3	Support for colour separation when using dvips
concmath	fonts3	Concrete math fonts derived from Computer Modern math fonts using parameters from Concrete text fonts. A LaTeX package providing the necessary font definition code is included.
concrete	fonts3	Concrete fonts

Catalogue of Packages *continued*

Package	*Collection*	*Description*
context	generic3	ppchtex is a package that can be used to typeset chemical formulas. The package is a separate module of the context macro package for TEX (context is a full featured, parameter driven macro package, which fully supports advanced interactive documents).
count1to	latex3	Setting `\count1` to `\count9`
croatian	lang3	Fonts for typesetting Croatian scripts
crosswrd	latex3	Macros for typesetting crossword puzzles.
csfonts	fonts2	Czech/Slovak-tuned METAFONT Computer Modern fonts.
cslatex	latex	LATEX support for Czech/Slovak typesetting.
csplain	plain2	Plain TEX support for Czech/Slovak typesetting.
curves	graphics3	Draws curves in the LATEX picture environment using parabolas between points with continuous slope at points. Equivalent to technical pens with compasses and French curves.
custom-bib	latex2	Package generating customized BIBTEX bibliography styles from a generic file using docstrip.
dancers	fonts3	Font for the Sherlock Holmes 'Dancing Men'
dates	latex3	Macros for parsing date strings.
deleq	latex3	Provides a more flexible numbering of equations, subequations, and 'recycled' equations, including 'partial' equation numbers ('3a', '3b' etc.).
devanagari	lang3	Fonts for typesetting Devanagari
dialogl	latex3	Macros for constructing interactive LATEX scripts
dinbrief	latex3	German letter DIN style.
doc1	texlive1	Basic documentation for **TEX Live**.
doc2	texlive2	Recommended documentation for **TEX Live**.
dotseqn	latex3	Flush left equations with dotted letters to the numbers
draftcopy	latex3	Places the word DRAFT (or other words) in light grey diagonally across the background (or at the bottom) of each (or selected) pages of the document.
dratex	graphics3	General drawing macros entirely in TEX
dropping	latex3	A LATEX 2_ε macro for dropping the first character(s) (or word(s)) of a paragraph. This is an extension of the LATEX 2.09 package dropcaps. This package automatically takes care of finding the font name.
dtk	latex3	Macros for the DANTE publication.
duerer	fonts3	Computer Duerer fonts.
dvips	generic1	Tom Rokicki's dvi to PostScript driver
dvipsbas	dvips1	Basic support files for dvips.
ean	generic3	Font for making EAN barcodes.
easy	latex3	Macros for simplying the writing of equations.
ec	fonts2	The ec fonts support the complete LATEX T1 encoding, as defined at the 1990 TUG conference hold at Cork/Ireland. They are intended to be as stable as the cm fonts are, i.e., there shall be no more changes to the tfm files. The ec fonts also contain a Text Companion Symbol font, called tc, featuring many useful characters needed in typesetting, for example oldstyle digits, currency symbols (including the newly created Euro symbol), the permille sign, copyright, trade mark and servicemark as well as a copyleft sign, and many others. Recent releases of LATEX 2_ε support the ec fonts. The dc fonts, which were termed as preliminary versions, will dissappear from the archives.
ecc	fonts3	'European' (T1 encoded) version of the Concrete fonts.
edmac	plain3	A macro package for typesetting scholarly critical editions
eepic	graphics2	A set of TEX macros for LATEX implementing several extensions to EPIC and the LATEX picture drawing environment, including the drawing of lines at any slope, the drawing of circles in any radii, and the drawing of dotted and dashed lines much faster with much less TEX memory, and providing several new commands for drawing ellipses, arcs, splines, and filled circles and ellipses.

Catalogue of Packages *continued*

Package	*Collection*	*Description*
eiad	latex3	Macros and EIAD fonts.
elsevier	latex3	Preprint style for Elsevier Science journals
elvish	fonts3	Font for typesetting Tolkien Elvish script
endfloat	latex3	Place all figures on pages by themselves at the end of the document with markers like "[Figure 3 about here]" appearing in the text (by default) near to where the figure (or table) would normally have occurred.
engwar	fonts3	Font for typesetting Tolkien Engwar script, by Michael Urban
envbig	latex3	Printing addresses on envelopes
envlab	latex3	A LaTeX 2ε package for producing mailing envelopes and labels, including barcodes and address formatting according to the US Postal Service rules. Redefines the standard `\makelabels` command of the LaTeX 2ε letter documentclass.
eplain	formats2	Simple but powerful extended version of the plain format, adding support for bibliographies, tables of contents, enumerated lists, verbatim input of files, numbered equations, tables, two-column output, footnotes and commutative diagrams.
epslatex	doc2	An extensive document which explains how to use Encapsulated PostScript (EPS) files in LaTeX 2ε documents. Includes explanations of Bounding Boxes, and more.
eqname	latex3	Style for different equation numbering.
eqnarray	latex3	More generalised equation arrays with numbering.
euler	latex3	Provides a setup for using the AMS Euler family of fonts for math in LaTeX documents. "The underlying philosophy of Zapf's Euler design was to capture the flavor of mathematics as it might be written by a mathematician with excellent handwriting." [concrete-tug] The euler package is based on Knuth's macros for the book "Concrete Mathematics". The text fonts for the Concrete book are provided by the beton package.
everysel	latex3	LaTeX package which provides hooks into `\selectfont`.
everyshi	latex3	Hooks for taking action at every `\shipout`
exam	latex3	Package for typesetting exam scripts.
exams	latex3	Exam questions can be multiple choice or free form long/short answer questions. Options include the typesetting of the exam itself, an exam showing all the answers and a collection of questions and answers. Questions can be parametrized. Use of a random generator provides for automatic shuffling of multiple choice items.
expdlist	latex3	Expanded description environments
export	latex3	This package allows the user to export/import the values of LaTeX registers (counters, rigid and rubber lengths only). It is definitely NOT for faint-hearted users.
fancyhdr	latex3	Support for sophisticated control of page headers and footers in LaTeX 2ε. It supersedes fancyheadings.
fancyheadings	latex2	Better control over page headers and footers in LaTeX. This is an up-to-date version for LaTeX. For LaTeX 2ε it is now called fancyhdr.
fancyvrb	latex2	Sophisticated handling of verbatim text, to write it out, read it in, and typeset it
fax	latex3	Document class for preparing faxes.
fc	fonts3	Fonts for African languages, complementary to Computer Modern.
feynmf	graphics3	Macros and fonts for creating Feynman (and other) diagrams.
float	latex2	Improves the interface for defining floating objects such as figures and tables. Introduces the boxed float and the ruled float. You can define your own floats and improve the behaviour of the old ones. Also incorporates the H option of the superseded here package. You can select this as automatic default with `\floatplacement{figure}{H}`.
floatfig	latex3	Allows text to be wrapped around figures.

Catalogue of Packages *continued*

Package	*Collection*	*Description*
floatflt	latex3	Float text around figures and tables which do not span the full width of a page. This is an improved version of floatfig. It is more or less similar to floatingtable. The tables/figures can be set left/right or alternating on even/odd pages. Works with the multicol package. Doesn't work well in the neighbourhood of list environments unless you change your LaTeX document.
fncychap	latex3	This package provides six predefined chapter headings. Each can be modified using a set of simple commands. Optionally one can modify the formatting routines in order to create additional chapter headings. This package was previously known as FancyChapter.
foiltex	latex3	A LaTeX 2_ε class for overhead transparencies. Can be used with fancybox to place a variety of borders around the slides.
fontinst	latex3	TeX macros for converting Adobe Font Metric files to TeX metric and virtual font format
fontmisc	fonts1	Miscellaneous METAFONT input files
fontname	texlive1	Karl Berry's scheme for naming fonts in TeX
fonts	latex1	Extra line and circle fonts for LaTeX
footnote	latex3	More sophisticated footnotes.
footnpag	latex3	Allows footnotes on individual pages to be numbered from 1, rather than being numbered sequentially through the document.
formats	texlive2	Prebuilt TeX format and METAFONT base files
fp	latex3	Provides an extensive collection of arithmetic operations for fixed point real numbers of high precision.
french	lang2	Style for French typography
fribrief	latex3	A LaTeX class for writing letters.
fundus	latex3	Providing LaTeX access to various font families.
futhark	fonts3	Fonts for the Older Futhark script
g-brief	latex3	A document class for LaTeX 2_ε. Serves for formatting formless letters in german or english language.
gb4e	latex3	Government Binding styles.
general	doc1	Useful general documentation.
genmisc	generic3	Miscellaneous small files for all formats, specific to the **TeX Live** CD-ROM.
geometry	latex3	A package which allows LaTeX 2_ε users to customise page layout (page sizes) using an easy and flexible user interface. You can specify `\geometry{body={6.5in,8.75in}, top=1.2in, left=2cm, nohead}`. This is an update of the now superseded pagesize package.
german	latex2	Style for German typography.
germbib	bibtex2	German variants of standard BibTeX styles.
go	fonts3	Fonts and macros for typesetting go games.
gothic	fonts3	Gothic and ornamental initial fonts by Yannis Haralambous.
graphics	latex2	The primary LaTeX package for the support of the inclusion of graphics generally produced with other tools. This package aims to give a consistent interface to including the file types that are understood by your printer driver.
gray	fonts3	Fonts for gray scales
hands	fonts3	Pointing hand fonts
harpoon	latex3	Extra harpoons, using the graphics package
harvard	bibtex2	The Harvard bibliography style family.
harvmac	plain3	Paul Ginsparg's Harvard macros for scientific articles
hh	latex3	Fancy boxing effects
histogr	latex3	Drawing histograms with the LaTeX picture environment.
hppa1.1-hpux10.20	systems1	System binaries HP running hpux10.20.
hppa1.1-hpux9.05	systems1	System binaries HP running hpux9.05.
html	doc2	Various TeX documentation converted to HTML

Catalogue of Packages *continued*

Package	*Collection*	*Description*
hyper	latex3	Redefines LaTeX cross-referencing commands to insert `\special` commands for HyperTeX dvi viewers
hyperref	latex3	Redefines LaTeX cross-referencing commands to insert `\special` commands for HyperTeX dvi viewers, or translation to Acrobat.
hyphen	lang1	Collection of hyphenation patterns.
i386-linux	systems1	System binaries for Intel machines running Linux.
i586-freebsd2.2	systems1	System binaries for Intel PC running FreeBSD 2.2.
i686-linux	systems1	System binaries for Intel Pentium Pro running Linux.
ieeepes	latex3	Allows typesetting of transactions, as well as discussions and closures, for the IEEE Power Engineering Society Transactions journals.
ifacmtg	latex3	Elsevier Science preprint style for IFAC meetings.
indxcite	latex3	A package to automatically generate an Author Index based on citations made using BibTeX. It requires the use of the harvard and index packages and LaTeX 2ε.
info	doc2	Documentation in GNU info form
inputenc	latex3	Controlling inpout encoding
ipa	latex3	Style for using International Phonetic Alphabet fonts
isostds	latex3	Class and package files for typesetting ISO International Standard documents. Several standard documents have been printed by ISO from camera-ready copy prepared using LaTeX and these files. One set of files is for generic ISO typesetting and the other is an extension set of packages for typesetting ISO 10303 standards.
jadetex	latex3	Macro package on top of LaTeX to typeset TeX output of Jade DSSSL implementation.
jknappen	latex2	Miscellaneous macros, mostly for making use of extra fonts, by Jörg Knappen.
jsmisc	plain3	Miscellaneous useful macros by Joachim Schrod.
jura	latex3	A document class for German law students.
knuth	doc3	Knuth's own documentation, including the TeXbook and the METAFONTbook
koma-script	latex3	A replacement for the article/report/book classes with emphasis on European rules of typography and paper formats as laid down by Jan Tschichold.
kuvio	graphics3	Drawing macros and fonts for diagrams.
labels	latex3	Support for printing sheets of sticky labels (but could also be used for business cards). The number of rows and columns of labels, and their size, can be changed.
lamstex	formats3	A merge of the best in AMSTeX and LaTeX
lastpage	latex3	Reference the number of pages in your LaTeX document (as in a page footer that says: Page N of M).
lgreek	latex3	Macros for using Silvio Levy's Greek fonts
lineno	latex3	Adds line numbers to selected paragraphs with reference possible through the LaTeX `\ref` and `\pageref` cross reference mechanism. Version 2.00 supports numbering of one in five lines and switching the line numbers from the left to the right side of the page in twoside mode.
listings	latex3	Package for pretty-printed program listings, with support for a range of languages.
localloc	latex3	Macros for localizing TeX register allocations
logic	fonts3	METAFONT font for drawing logic diagrams.
lollipop	formats3	A new generation format
lshort	latex1	Short introduction to LaTeX.
ltablex	latex3	Modifies the tabularx environment to combine the features of the tabularx package (auto-sized columns in a fixed width table) with those of the longtable package (multi-page tables).
ltxdoc	latex1	Class for documented LaTeX 2ε classes.
ltxmisc	latex2	Miscellaneous LaTeX styles.

Catalogue of Packages *continued*

Package	*Collection*	*Description*
lw35ps	fonts2	Font metrics and LaTeX font description files for standard 35 PostScript fonts.
ly1	latex3	Support for LY1 LaTeX encoding, i.e. the Y&Y texnansi encoding.
mab-nextstep3	systems1	System binaries for Next boxes running NextStep 3.
mailing	latex3	Macros for mail merging
makeindex	texlive1	Documentation for the MakeIndex program.
malvern	fonts3	A new sans-serif font family
mapcodes	latex3	Support for multiple character sets and encodings.
maple	latex3	Styles and examples for the MAPLE newsletter.
mathcomp	latex2	A package which provides access to some interesting characters of the Text Companion fonts (TS1 encoding) in math mode.
mcite	latex3	Support for collapsing multiple citations into one, as customary in physics journals
mdwtools	latex3	Miscellaneous tools by Mark Wooding, including support for `@`, a doafter command, footnotes, mathenv for various alignment in maths, list handling, trivial maths oddments, rewrite of LaTeX's `tabular` and `array` environments, verbatim handling, and syntax diagrams.
mff	latex3	A package to provide something similar to 'multiple master' fonts, but using METAFONT; you specify a font by a set of METAFONT parameters, and TeX makes up a .mf file to generate the required font; this package is not integrated with NFSS (or MakeTeXTFM) yet fun.
mflogo	latex1	LaTeX package and font definition file to access the Knuthian 'logo' fonts described in 'The METAFONTbook' and the METAFONT and MetaPost logos in LaTeX documents.
mfmisc	fonts1	Small support files for METAFONT.
mfnfss	latex3	Font description files for extra fonts like yinit and ygoth
mfpic	graphics3	Macros which generate METAFONT code for drawing pictures.
mftoeps	fonts3	a METAFONT package for generating (encapsulated PostScript) files readable by CorelDRAW!, Adobe Illustrator and Fontographer. METAFONT writes PostScript code to a LOG-file, and from the LOG-file the code can be extracted by either TeX or AWK. DOS batch files, TeX source, and AWK source are placed in the subdirectory PROGS.
midnight	generic3	A set of useful macro tools
minitoc	latex3	Table of contents per chapter.
mips-irix4.0.5	systems1	System binaries for SGI box running Irix 4.0.5.
mips-irix5.3	systems1	System binaries for SGI box running Irix 5.3.
mips-irix6.3	systems1	System binaries for SGI box running Irix 6.3.
mips-ultrix4.4	systems1	System binaries for MIPS machine running Ultrix 4.4.
misc	latex3	Miscellaneous small macro files for LaTeX.
misc209	latex2	Miscellaneous small macro files for LaTeX2.09.
mnras	plain3	Styles for the Monthly Notices of the Royal Astronomical Society.
monster	latex3	Towards a more rational and modular LaTeX, by Matt Swift; a set of powerful tools
moreverb	latex3	A verbatim mode that can handle TABs properly, can number lines, can number lines in an included file, can produce boxed verbatims, etc.
mpbase	metapost1	Basic MetaPost support files.
mpfnmark	latex3	A package which provides the command `\mpfootnotemark`, which can be used in the same way as `\footnotemark`. The difference between these two macros is that within minipage environments the latter uses the standard footnote marker style (defined by `\thefootnote`), while the new command uses the minipage footnote marker style (defined by `\thempfootnote`).
mslapa	latex3	LaTeX and BibTeX style files for a respectably close approximation to APA (American Psychological Association) citation and reference style.
mtbe	plain3	Examples from *Mathematical TeX by Example* by Arvind Borde
musictex	generic3	Typesetting music with TeX.
musixtex	generic3	Extended MusicTeX, with better slurs

Catalogue of Packages *continued*

Package	*Collection*	*Description*
myletter	latex3	Another letter package.
nassflow	latex3	Drawing Nassi-Schneidermann diagrams.
natbib	bibtex2	A bibliography style that handles author-year and numbered references.
newalg	latex3	Format algorithms like Cormen, Leiserson and Rivest.
newsletr	plain3	Macros for making newsletters
newthm	latex3	A modified version of the theorem-style which provides generation of lists of theorems.
niceframe	latex3	Support for fancy framing of pages
nomencl	latex3	Nomenclature package for producing lists of symbols using the capabilities of the MakeIndex program.
ntgclass	latex2	Versions of the standard LaTeX article and report classes, rewritten to reflect a more European design, by the Dutch TeX Users Group
numline	latex3	Macros for numbering lines.
objectz	latex3	Macros for typesetting Object Z
oca	fonts3	OCR font
ocr-a	fonts3	Fonts for OCR-A
ocr-b	fonts3	Fonts for OCR-A
ogham	fonts3	Fonts for typesetting Ogham script
ogonek	latex3	Support for Polish typography and the ogonek
oldstyle	latex3	Font information needed to load the `cmmi` and `cmmib` fonts for use to produce oldstyle numbers
osmanian	fonts3	Osmanian fonts by Alan Stanier for writing Somali
ot2cyr	fonts2	Macros to use to the OT2 Cyrillic encoding
overword	latex3	This package provides two macros which can be used as building blocks for the parsing of text. For an example of their use, see the calendar package.
pandora	fonts3	The Pandora font family
paper	latex3	A class derived from article, tuned for producing papers for journals. Introduces new layout options and font commands for sections/parts. Defines a new keywords environment, and subtitle and institution commands for the title section. New commands for revisions. And more.
parallel	latex3	Provides a parallel environment which allows two columns of text to be typeset. Useful for typesetting two languages side-by-side.
patch	latex3	Macros for package management.
pb-diagram	latex3	Diagram package, using LAMSTeX fonts, by Paul Burchard.
pdcmac	plain3	Damian Cugley's macro tools.
pdf	doc2	Documentation in PDF format.
pdftex	plain2	Macro packages for variant TeX which writes PDF format.
phonetic	fonts3	METAFONT Phonetic fonts, based on Computer Modern
physe	formats3	PHYSE format
phyzzx	formats3	A TeX format for physicists
picinpar	latex3	Insert pictures into paragraphs. (NOTE: Piet van Oostrum does not recommend this package. Picins is recommended instead.)
pictex	graphics2	Picture drawing macros for LaTeX.
piff	latex3	Macro tools by Mike Piff
plainmisc	plain1	Miscellaneous useful macros for plain TeX
plaintex	plain1	Basic Plain TeX macros
plfonts	fonts3	Computer Modern variant fonts for Polish
plgraph	generic3	LaTeX graphics package with wrapper to allow it to be used with generic plain TeX
pmgraph	latex3	A set of extensions to LaTeX picture environment, including a wider range of vectors, and a lot more box frame styles.
poligraf	generic3	Page preparation for prepress, color separation, crop-marks, color and gray scale bars, booklet preparation, etc.
prelim2e	latex3	Allows the marking of preliminary versions of a document

Catalogue of Packages *continued*

Package	*Collection*	*Description*
prettyref	latex3	Additional functionality to LaTeX 2ε label–reference mechanism. It allows the author to "preformat" all types of labels.
progkeys	latex3	The file 'programs.sty' is intended to allow a parameterized way of typesetting programs with TeX/LaTeX commands inside. The file 'keywords.sty' allows use and define sets of keywords that will be typeset with different fonts, according to the wish of the user.
program	latex3	Typesetting programs and algorithms
proofs	latex3	Macros for building proof trees.
psfonts	fonts2	PostScript fonts for use with TeX and LaTeX.
psfrag	graphics2	A set of macros and a PostScript header which allows LaTeX constructions (equations, picture environments, etc.) to be precisely superimposed over Encapsulated PostScript figures. The user can use his/her favorite drawing tool to create an EPS figure, placing simple text "tags" where each replacement is to be placed. PSfrag will automatically remove those tags from the figure and replace them with the LaTeX construction that the user specifies, properly aligned, scaled, and/or rotated.
psizzl	formats3	A TeX format from SLAC
pslatex	latex2	A small package that makes LaTeX default to 'standard' PostScript fonts. It is basically a merger of the times and mathptm styles from the psnfss suite of packages. You must have installed standard LaTeX and PSNFSS PostScript fonts to use this package. The main novel feature is that the pslatex package tries to compensate for the visual differences between the Adobe fonts by scaling Helvetica by 90%, and 'condensing' Courier (i.e. scaling horizontally) by 85%. The package is supplied with a (unix) shell file for a 'pslatex' command that allows standard LaTeX documents to be processed, without needing to edit the file.
psnfss	fonts2	Font definition files, macros and font metrics for common PostScript fonts
psnfssx	latex3	Extra styles and encodings for PS fonts, including Y&Y encoding support.
pspicture	latex2	Replacement for core LaTeX picture macros to use PostScript `\special` commands
pstricks	graphics2	An extensive collection of PostScript macros that is compatible with most TeX macro packages, including Plain TeX, LaTeX, AMS-TeX, and AMS-LaTeX. Included are macros for color, graphics, pie charts, rotation, trees and overlays. It has many special features, including: a wide variety of graphics (picture drawing) macros, with a flexible interface and with color support. There are macros for coloring or shading the cells of tables.
punk	fonts3	Donald Knuth's punk font
qobitree	graphics3	LaTeX macros for typesetting trees
qsymbols	latex3	For defining systematic mnemonic abbreviations, starting with ` for math symbols and \" for arrows, from the amssymb and stmaryrd packages
ragged2e	latex3	LaTeX package which defines new commands `\Centering`, `\RaggedLeft`, and `\RaggedRight` and new environments Center, FlushLeft, and FlushRight, which set ragged text and are easily configurable to allow hyphenation.
rcs	latex3	Use RCS (revision control system) tags in LaTeX documents.
realcalc	plain3	Macros for real arithmetic calculations.
refman	latex3	Variant report and article styles
revtex	latex2	Styles for American Physical Society, American Institute of Physics, and Optical Society of America. Only works in compatibility mode under LaTeX 2ε.
rlepsf	generic3	A macro package for use with epsf.tex which allows PostScript labels in an eps file to be replaced by TeX labels.
rotating	latex2	A package built on the standard LaTeX graphics package to perform all the different sorts of rotation one might like, including complete figures and tables and captions.
rotfloat	latex3	Rotate floats.

Catalogue of Packages *continued*

Package	*Collection*	*Description*
rplain	latex3	Redefines the 'plain' pagestyle. The page numbers are now in the lower right corner.
rs6000-aix3.2.5	systems1	System binaries for RS6000 running AIX 3.2.5.
rs6000-aix4.1.1	systems1	System binaries for RS6000 running AIX 4.1.1.
sauter	fonts3	Extensions to the CM fonts, providing a parameterization scheme to build fonts at true design sizes
scale	latex3	A package to scale a document by $\sqrt{2}$. This is useful if you are preparing a document on e.g. A5 paper and want to print on A4 paper to achieve a better resolution.
script	latex3	Variant report / book styles
semantic	latex3	Eases the typesetting of notation of semantics and compilers. Includes T-diagrams, various derivation symbols and inference threes.
seminar	latex2	Produce overhead slides (transparencies) with bells and whistles.
setspace	latex3	Provides commands and environments for doing double and one-and-a-half line spacing based on pt size. If a different spacing is required then the `\setstretch{baselinestretch}` command is supported. The spacing environment takes one argument which is the baselinestretch to use, e.g., `\begin{spacing}{2.5}`.
shadbox	latex3	A tool to shade the background of any box – text, figure, table etc. – using Plain (LA)TEX.
shadethm	latex3	Package that allows declarations of the form `\newshadetheorem{thm}{Theorem}` or `\newshadetheorem{}[]{}` or `\newshadetheorem{}{}[]` to produce shaded boxes from the usual command `\begin{theorem}` ... `\end{theorem}`. The color package is required.
showlabels	latex3	Show label commands in the margin.
siam	generic3	Styles for SIAM publications
siggraph	latex3	Document class for formatting papers according to the specifications for submission to the annual ACM Siggraph conference
slidenotes	latex3	A class package for the easy production of a slide collection with annotations. Builds on the report style (or variants).
smallcap	latex3	Support for all 4 shapes of Small caps in DC1.3 where SC becomes a family, rather than a shape (`\scshape` is replaced by `\scfamily`). Thus you can write `\bf\scfamily\slshape` to get small caps bold slanted.
songbook	latex3	Package for typesetting song lyrics.
sparc-linux	systems1	System binaries for Sparc running Linux.
sparc-solaris2.4	systems1	System binaries for Sparc running Solaris 2.4.
sparc-solaris2.5	systems1	System binaries for Sun Sparc running Solaris 2.5.
sparc-sunos4.1.3	systems1	System binaries for Sparc running SunOS 4.1.3.
sprite	graphics3	Macros to set bitmaps with TEX
ssqquote	latex3	LATEX package and font definition file to access the 'cmssq' fonts, i.e. Computer Modern Sans Serif Quotation Style. The LATEX package also defines a `chapterquotes` environment as an example application.
startex	formats3	A TEX format designed to help students write short reports and essays. It provides the user with a suitable set of commands for such a task. It is also more robust than plain TEX and LATEX.
stmaryrd	fonts2	St Mary Road symbols for functional programming.
subeqn	latex3	Package for subequation numbering.
subeqnarray	latex3	Equation array with sub numbering.
subfigure	latex3	Figures divided into subfigures.
supertabular	latex3	A multi-page tables package.
swift	latex3	Miscellaneous macros by Matt Swift.
tap	plain3	An advanced table package.
taylor	graphics3	Diagram macros by Paul Taylor.
tbe	plain3	Examples from Arvind Borde's *TEX by Example*
tengwar	fonts3	Font for typesetting Tolkien Tengwar script, by Michael Urban

Catalogue of Packages *continued*

Package	*Collection*	*Description*
tex-ps	generic3	Various extra support file for dvips.
texdraw	graphics3	Graphical macros, using embedded PostScript.
texip	formats3	Macros from *TEX in Practice*
texlive	texlive1	Basic material for **TEX Live**.
text1	formats3	TEX format from the University of Washington
textcomp	latex3	Supports the Text Companion fonts which provide many text symbols (such as baht, bullet, copyright, musicalnote, onequarter, section, and yen) in the TS1 encoding.
textfit	latex3	Package to support fitting of text to a given width or height by scaling the font
textmerg	latex3	Merge text in TEX and LATEX. Useful, for example, in mail merge.
thesis	latex3	A class for producing a thesis based on the report class for a more European and more flexible look. Supports options like noindent, noitemization, headline, nocenter, crosshair, and chapterbib.
timesht	latex3	Package for typesetting time sheets.
tipa	fonts3	Fonts and macros for IPA phonetics characters.
tools	latex2	Standard LATEX 2_ε tools, for extended tabular, verbatim and theorem support
tracking	latex3	Automatically adjust spaces between symbols in words or phrases to fit them into a specified length. Any chain of symbols (including spaces) in the current font may be treated.
treesvr	latex3	Tree macros.
treetex	plain3	Allows the automatic layout of n-ary trees with arbitrary node sizes in LATEX, using an external C program to do much of the hard work.
tugboat	generic2	Macros for TUGboat articles (plain and LATEX).
type1cm	latex3	A package that removes the restriction when using scalable versions of the cm fonts (Type1 Bakoma, or versions from BSR/Y&Y, or True Type versions from Kinch, PCTeX etc.) where LATEX restricts the cm fonts to discrete sizes.
typehtml	latex3	Typeset HTML (i.e., World Wide Web documents) directly from LATEX. Can handle almost all of HTML2, and most of the math fragment of the draft HTML3.
uaclasses	latex3	This package provides a LATEX 2_ε document class named 'ua-thesis' for typesetting theses and dissertations in the official format required by the University of Arizona. Moreover, there is a fully compatible alternative document class 'my-thesis' for private "nice" copies of the dissertation, and the respective title pages are available as separate packages to work with "any" document class.
ucthesis	latex3	A modified version of the standard LATEX REPORT style that is accepted for use with University of California PhD dissertations and Masters theses.
ulsy	latex3	Extra mathematical characters
umlaute	latex3	An interface to inputenc for using alternate input encodings
umrand	fonts3	Package for page frames.
underlin	latex3	Package for underlining. Be advised that underlining is considered bad style in typesetting. See also ulem which is a specific package for LATEX.
useful	doc2	Useful documentation; various LATEX guides, FAQ, fontname docs, etc.
utthesis	latex3	Produces a thesis that meets the requirements of the Graduate School of The University of Texas at Austin :-)
uwthesis	latex3	University of Washington thesis
vdm	latex3	Typesetting VDM schemas
vector	latex3	Macros for more convenient representation of vectors in LATEX 2_ε, both symbolically and as implicit or explicit rows/columns of elements
vertex	plain3	Styles for economics working papers and journals
vita	latex3	This class provides necessary macros to prepare your Curriculum Vitae or Resume.
vrb	latex3	Verbatim macros via plain TEX

Catalogue of Packages *continued*

Package	*Collection*	*Description*
vrsion	latex3	Defines a command which produces a version number in the .dvi-file when LaTeX is run.
wasy	fonts3	The wasy fonts (Waldis symbol fonts)
wasysym	latex2	Makes some additional characters available that come from the wasy fonts (Waldis symbol fonts). These fonts are not automatically included in NFSS2/LaTeX 2_ε since they take up important space and aren't necessary if one makes use of the packages amsfonts or amssymb. Symbols include: join, box, diamond, leadsto, sqsubset, lhd, rhd, apprle, ocircle, invneg, logof, varint, male, female, phone, clock, lightning, pointer, sun, bell, permil, smiley, various electrical symbols, shapes, music notes, circles, signs, astronomy, etc.
williams	latex3	Miscellaneous macros by Peter Williams.
win32	systems1	System binaries for Windows 32.
wnri	fonts3	METAFONT fonts for Old English, Indic languages in transcription, and American Indian languages.
wsuipa	fonts2	Washington State University IPA phonetic fonts
xymtex	latex3	Typesetting chemical structures.
xypic	graphics2	Sophisticated macros and fonts, originally designed for commutative diagrams, but with general applicability.
yhmath	latex3	Extended maths fonts for LaTeX.
youngtab	latex3	A package for typesetting Young-Tableaux mathematical symbols for the representations of groups, providing two macros, `\yng(#1)` and `\young(#1)` to generate the whole Young-Tableaux.
ytex	formats3	Macro package developed at MIT.
zed-csp	latex3	Typesetting Z and CSP format specifications

◇ Sebastian Rahtz
7 Stratfield Road
Oxford OX2 7BG
UK
s.rahtz@elsevier.co.uk

◇ Michel Goossens
CERN
CH-211 Geneva 23
Switzerland
m.goossens@cern.ch

Fonts

Virtual Fonts, Virtuous Fonts

Alan Hoenig

Abstract

Virtual fonts allow us to use all digital fonts with TEX, even non-TEX ones, and do much more for us. What are virtual fonts? Several projects grant us necessary experience with them.

This article comprises somewhat less than half of the similarly named chapter which will appear in the book *TEX Unbound: LATEX and TEX Strategies for Fonts, Graphics, and More*, by Alan Hoenig, to be published in 1997 by Oxford University Press. This excerpt has been revised so that it may be printed using the standard suite of TEX fonts; the original depends heavily on PostScript fonts and the author's style file for its typesetting. Consequently, some displays could not be included. For any questions or comments, please contact the author at `ajhjj@cunyvm.cuny.edu`.

When talking about computers, we use the adjective "virtual" to describe a thing that behaves like something else. Virtual disks are really memory blocks which simulate hard disks, while virtual memory uses a disk to mimic a computer's memory. A virtual font looks to TEX like any other font, but it really is pieced together from other fonts or collections of typographic elements. It may be

- a composite of several different fonts somehow mixed together (in a special way, according to precise rules);
- a single font whose characters are (for very good reasons) scrambled in some new order;
- a collection of built-up characters, each constructed from several components (like accented letters often are), which behaves like a font;
- individual horizontal or vertical rules each of which is treated as a character in a font;
- a collection of text, graphics, or PostScript files, each of which is treated as a single character within the virtual font;
- a conglomerate of all (or some) of the above.

This article explores virtual fonts, considers some occasions that need them, and provides procedures for constructing them. It's messy constructing virtual fonts by hand, but a few freely available resources make it easy.

The box (not included here — sorry) lists some virtual font projects. For most people, the only application of virtual fonts may be to perform the proper installation of outline fonts (PostScript fonts) for use by TEX. (We will use the term *installation* to describe the entire process of making fonts usable by TEX.) Many tasks difficult or impossible to accomplish with macros become trivial when implemented via virtual fonts.

The next few sections explain the concept of virtual font, together with the related concepts of font tables and encoding tables, in detail. Thereafter, we will provide discussion and procedures for implementing most of the applications in the list above.

1 The virtual font concept

Let's begin by journeying to a different planet, one on which a system like TEX has been developed, but on which all languages contain only two distinct characters, which we can call 'e' and 'f', together with the double-f ligature 'ff'. A close examination of a font on this hypothetical planet makes it easier to understand the kinds of problems arising in real, terrestrial fonts, and how virtual fonts can solve them.

A table listing the characters of any TEX font would contain only three characters.

0	1	2
e	f	ff

These three characters have been numbered using the usual computer science convention which starts with 0. These numeric labels serve to identify the position in the font of each character.

These numeric positions also play an important role for TEX, for `dvi` files contain typesetting commands based *not* on the glyph name ('A', 'B', 'comma', or whatever) but on each numeric label. For any 'e' in the input file, the `dvi` file contains the instruction to typeset character 0 in the current font. The lowercase 'e' had better be in that position! This correspondence between character and character number is built into the TEX program, and that's true for both the distant planet and for ours.

Difficulties arise when we try to use a commercial font instead of Computer Modern. We will suppose that the commercial font we want contains three characters, but they are 'e', 'f', and '&'. To get the ligature, we need to purchase a separate font, which contains the 'ff' plus two other characters. A further difficulty surfaces when we examine the

layout of the fonts.

0	1	2
f	e	&

0	1	2
ff	%	$

The characters in the main font are in the wrong order, and this leads to disaster. To see why, let's select the commercial font, and now suppose we type `f`. TEX expects an 'f' to occupy position 1 of the font table, and so puts an instruction (in the `dvi` file) to typeset character 1. But character 1 in the non-TEX font is the glyph 'e', and that's what gets typeset — not the 'f' that we requested. Moreover, it does not appear that we can typeset the ff without an explicit call to the auxiliary font. Apparently, the input file will look different whether we typeset with the usual TEX fonts or with some other fonts, and this is unacceptable.

Virtual fonts have been created to deal with this (and other) exigencies. As far as an author is concerned, a virtual font is just another font. But it provides a mechanism whereby (behind the scenes), real fonts (*raw fonts*) can be combined so that the resulting *virtual font* conforms to the usual TEX conventions to eliminate any need for marking up the input file differently. In our example, a virtual font would

1. select the e and f from the main font, and reorder them in a TEX-acceptable way; and
2. include the ligature from the expert font in the last position in the table for the virtual font.

We call an auxiliary font containing ligatures and other special symbols an expert font. Furthermore, we'll follow the terrestrial convention of labelling raw fonts by appending '8a' or '8x' (expert) to them. So raw font `foo8a` and expert font `foo8x` come together in the virtual font `foo7t`. (The `fontname` convention due to Karl Berry dictates the conventions surrounding the notations 8a, 8x, and 7t. Explanations behind these conventions can be found in any CTAN archive or mirror in the `info/fontname` area.)

0	1	2
f	e	&

+

0	1	2
ff	%	$

⇒

0	1	2
e	f	ff

`foo8a` `foo8x` `foo7t`

Font `foo7t` uses selected characters from the two raw fonts, and orders this selection in a way meaningful to TEX. Not all the characters from raw fonts need be part of the final virtual font.

Up until virtual fonts, words containing accents suppressed TEX's hyphenation algorithm. We can define an accented letter in a virtual font to be equivalent to any other letter, so hyphenation proceeds unimpeded — yet another advantage of virtual fonts.

To be sure, this unrealistic, alien font provides a contrived example. But with real commercial fonts, these same problems — magnified because real fonts have so many more characters — need the practical solutions that virtual fonts provide.

2 Digital fonts and font tables

Font tables for the Computer Modern TEX fonts and for PostScript outline fonts contain a maximum of 256 positions; 256 is one of the magic numbers of computer science. See figure 1 for examples of real font tables. (For a slightly different representation, refer to the font tables beginning on page 427 of *The TEXbook*.) A casual glance reveals significant differences between the layouts for the two fonts. Each slot of the Computer Modern font is filled (up till character 127), whereas there are many unfilled slots in the PostScript fonts. Some characters, like the uppercase Greek letters or the ff, ffi, and ffl ligatures, do not appear anywhere in the PostScript font while certain PostScript characters appear nowhere in the Computer Modern layout. Other characters are in disparate positions. All the Scandinavian ligatures (æ, and so on) appear in the fourth row of the Computer Modern font table, but cluster together near the very end of the PostScript font table.

Positions in any font table are numbered starting from zero up to 255. We know that terrestrial TEX selects characters *not* by the character name but according to its position in the font table, so a command to typeset an 'A' is relayed as an instruction to

> typeset the character from the currently selected font that occupies position 65 in that font,

since that's the numeric label of the 'A' slot. (All character positions are given here in decimal notation. Computer scientists may be more comfortable with a character's octal position, which is why that information also appears in the tables.)

When using PostScript outline fonts, it's useful to be able to typeset in a 'fake font' — our virtual font — which looks real to TEX but is in fact an amalgam of one or more raw, component fonts. We arrange this virtual font so the characters in the virtual font are in the same order as in any other Computer Modern font. That way, macros will seldom have to be redefined for different fonts, a particularly important issue for mathematics typesetting.

Associated with a font table is the *font encoding vector* or just the *encoding vector* or *code page*. The encoding vector is the list of the character names in

0 ` 0	1 Δ 1	2 Θ 2	3 Λ 3	4 Ξ 4	5 Π 5	6 Σ 6	7 Υ 7
8 Φ 10	9 Ψ 11	10 Ω 12	11 ff 13	12 fi 14	13 fl 15	14 ffi 16	15 ffl 17
16 ı 20	17 ȷ 21	18 ˋ 22	19 ´ 23	20 ˘ 24	21 ˇ 25	22 ¯ 26	23 ˚ 27
24 ¸ 30	25 ß 31	26 æ 32	27 œ 33	28 ø 34	29 Æ 35	30 Œ 36	31 Ø 37
32 - 40	33 ! 41	34 ” 42	35 # 43	36 $ 44	37 % 45	38 & 46	39 ’ 47
40 (50	41) 51	42 * 52	43 + 53	44 , 54	45 - 55	46 . 56	47 / 57
48 0 60	49 1 61	50 2 62	51 3 63	52 4 64	53 5 65	54 6 66	55 7 67
56 8 70	57 9 71	58 : 72	59 ; 73	60 ¡ 74	61 = 75	62 ¿ 76	63 ? 77
64 @ 100	65 A 101	66 B 102	67 C 103	68 D 104	69 E 105	70 F 106	71 G 107
72 H 110	73 I 111	74 J 112	75 K 113	76 L 114	77 M 115	78 N 116	79 O 117
80 P 120	81 Q 121	82 R 122	83 S 123	84 T 124	85 U 125	86 V 126	87 W 127
88 X 130	89 Y 131	90 Z 132	91 [133	92 “ 134	93] 135	94 ^ 136	95 ˙ 137
96 ‘ 140	97 a 141	98 b 142	99 c 143	100 d 144	101 e 145	102 f 146	103 g 147
104 h 150	105 i 151	106 j 152	107 k 153	108 l 154	109 m 155	110 n 156	111 o 157
112 p 160	113 q 161	114 r 162	115 s 163	116 t 164	117 u 165	118 v 166	119 w 167
120 x 170	121 y 171	122 z 172	123 – 173	124 — 174	125 ˝ 175	126 ~ 176	127 ¨ 177

Figure 1: A font table for Computer Modern fonts (here, `cmb10`). The upright numbers in the upper left of each box give the character number using the usual decimal representation. The italic numbers in the lower right are the octal equivalents.

the order in which they occur in the font table. For a TEX font (figure 1), the encoding vector is the list beginning

```
Gamma, Delta, Theta, Lambda, Epsilon, Pi,
Sigma, Upsilon, Phi, Psi, Omega,
ff, fi, fl, ffi, ffl, dotlessi, ...
```

and so on. If we let '`.notdef`' designate a font position for which no character is defined, then for a PostScript font, the encoding vector is a list that begins

`.notdef, ..., .notdef` (32 times in all),
`space, exclam, quotedbl, numbersign, dollar,`
`...`

3 What comprises a virtual font?

TEX does not deal with any characters beyond the metrics associated with a font. It expects to find this information in a `tfm` file, and so each virtual font must be accompanied by a font metric file in the usual way. This file should be placed in a suitable place.

The details behind the construction of the virtual characters appear in the actual virtual font file, a file with the extension `vf`. There needs to be a place on a hard disk to store virtual fonts, in the same way that there are places for `tfm` files, format files, input files, and so on. The places have different names depending on whether your system is traditional or complies with the TDS standard.

The actual virtual font `vf` file contains fragments of `dvi` language that specify the way that a virtual character should be created. That means that a character in a virtual font can be anything that occurs in a `dvi` file. In theory, one virtual character can typeset an entire page or document! Typically, virtual characters are not so complex. In the alien planet example, the virtual font simply remapped characters (placed them in a different and more suitable order) and merged characters together from raw fonts.

4 What we will need; preparation

Firstly, virtual fonts are a feature of TEX3. In order to proceed, that version must be installed.

Many authors will be preparing documents for output on PostScript printing devices. Since TEX

only knows how to write `dvi` files, we will always need a dvi-to-PostScript converter. Frequently these programs require an auxiliary map file to "map" the long font names to the short file names that are all that some operating systems, notably MS-DOS and its relatives, can handle. Because it is freely available, and available for all computer platforms, we will usually refer to *dvips* and its map file `psfonts.map`. Each of its entries pairs a short, DOS-acceptable name for a *raw* font with its long, given font name. These short aliases are the names that we should use in the process of virtual font creation. For each short alias in the map file, there must be a `tfm` file under that name.

The map file may serve other functions. It may aid in the process of downloading (see below), and it may be where we specify certain types of transformations on a font.

At print time, how does the printer get the information about the shapes of the characters in the document? For bitmap fonts, it's the responsibility of the printer driver to include the bitmap information in the instructions it transmits to the printer. For scalable fonts, the situation is different. The outline information on all fonts must be transmitted to the printer, for it is the printer that ultimately converts the outline to raster form for printing. In most PostScript-compatible printers, descriptions of 35 or so common fonts, including Times Roman and Helvetica, are resident — built-in — to the printer. If you use other, non-resident fonts, you will need to *download* — transmit — this font information to the printer, and this downloading can be accomplished in different ways. It is also possible to include the font information in the PostScript version of the document.

5 The purpose of a simple installation

If we examine the font tables in this article, we see that the problem of constructing a virtual font from a PostScript font is not hopeless. Most characters are in the same positions, including all upper- and lowercase letters, digits, and much of the punctuation. We may divide the remaining characters in an outline font into two groups:

- special characters like fi, —, ¿, and the American quotation marks “ ” which are selected by TEX's ligature mechanism; and
- characters like æ, Œ, or ç which are invoked by control sequences or control words (here, `\ae`, `\OE`, and `\c{c}`).

(Actually, there's a third group — those characters present in `cmr10` but absent entirely from a standard Type1 font, such as the ligatures ff, ffi, and ffl. We'll see later how to deal with these.) We would like to make sure we have access to *these* members of a font **without** having to change the rules by which we create our source documents. Actually, just in case an author has been silly and used a non-standard convention to typeset a symbol (such as getting ¿ by typing `\char62` or `\symbol{62}` rather than `?‘`), we would like the layout of the virtual font to adhere as closely as possible to the original TEX font layout.

The ligatures of the first group can be handled in a non-virtual way by adjusting the font metric files so TEX plucks the ligature from the proper font position; no remapping is necessary. This requires a modification of the `tfm` only.

Characters accessed by TEX commands present more of a challenge. The definition for each such command relies upon being able to locate special characters by their position in the font table. TEX therefore expects œ to be character 27 in a font, since that's where it is in the Computer Modern family. When constructing ç, it expects the cedilla to be in position 24 for the same reason. Typically, though, these characters do not appear in those positions in the raw PostScript font (œ and cedilla occupy positions 250 and 203). Macros could be redefined, but it's a bad idea to have macro definitions depend on the current font. We require our virtual font utility to reorder — to remap — these characters in the font. For example, virtual character 27 consists of the raw character 250. That way, when the virtual font is the current font, `\oe` will correctly typeset the œ glyph.

The `afm2tfm` utility (part of *dvips*) is an excellent tool for creating this elementary kind of virtual font — a font consisting of the remapping of the characters in a single raw font. Because the source for this program has been made available, `afm2tfm` has been ported to every significant computer architecture, and executable binaries are freely available from friends or software archives (the same applies to *dvips* itself). But `afm2tfm` suffers from several disabilities: it can't create a virtual font out of more than one raw file, it can't create the `fd` font descriptors that LATEX now uses, and it doesn't mimic the original TEX font layout as closely as it might. Nevertheless, simple installations are so common that it is important to detail this process precisely.

The material in figure 2 summarizes the procedure to follow to use `afm2tfm` to create virtual files from outline fonts. We use this procedure whenever this simple manipulation is sufficient for our needs. (More complicated finagling is best carried

Simple Font Installation with `afm2tfm`

We use `afm2tfm` to install a font whenever a single raw font is the sole component of a virtual font. Here are the steps in the procedure, assuming we begin with font files `foo8a.afm` and one of `foo8a.pfb`, `foo8a.pfa`, or `foo8a` (depending on the computer system and the way the fonts are distributed). From these, we follow these steps to create a virtual font `foo7t`.

1. Issue the command

   ```
   afm2tfm foo8a.afm -v foo7t.vpl foo8a.tfm >tmp
   ```

   at the prompt.
2. Issue the command

   ```
   vptovf foo7t.vpl foo7t.vf foo7t.tfm
   ```

   at the prompt.
3. The file `tmp` contains a single line like

   ```
   foo8a FullNameOfTheFont
   ```

   Add this line to the end of the file `psfonts.map`. If you want `dvips` to download the file automatically, revise the line to look like

   ```
   foo8a FullNameOfTheFont </psfonts/foo8a.pfb
   ```
4. Move the two `.tfm` files, a single `.vf` file, and the original font files to their proper directories. Make sure also that the revised version of `psfonts.map` is where it belongs.

Figure 2: Creating virtual fonts with `afm2tfm`.

out with *fontinst*; see below.) This process involves using or creating several file types. If an outline font is `psfont`, that means the distribution diskette should include `psfont.afm` and `psfont.pfb`. It is necessary to rename the file name `psfont8a`, and from these we will be generating files `psfont7t.vpl`, `psfont7t.tfm`, and `psfont7t.vf`, the virtual file. We also generate a font metric file for the "raw" PostScript file `psfont8a.tfm`.

The program `afm2tfm` can also create pseudo-small caps fonts and other fonts which have undergone simple geometric transformations, like slanting or extension. Check the documentation to learn how.

Once we've created the virtual font and placed all the files where they belong, we access any virtual file just as if it were a normal TEX font (which it is). For example, we could declare

```
\font\foo=psfont7t at 10.5pt
```

in a `plain` TEX document and use it via the command `\foo` which has become a font changing command like `\it` or `\tt`. Although we never again refer to the raw font file explicitly, TEX does. Behind the scenes, whenever a TEX device driver resolves the meaning of a virtual font, it refers to the component raw fonts. The raw fonts must be present on our system.

6 Introduction to *fontinst*

The *fontinst* package, by Alan Jeffrey, does everything `afm2tfm` does and more. It can create a virtual font from several raw fonts, for example, and it automatically produces an auxiliary `fd` file used by LATEX's NFSS to select the font. The *fontinst* package is written entirely in TEX, and TEXegetes will enjoy perusing `fontinst.sty` to watch TEX do things it was never intended for. Writing it in the TEX language insures *fontinst* runs on every platform that TEX does. You can retrieve *fontinst* from any CTAN archive, under `fonts/utils`. The discussion here supplements `fontinst.tex`, the documentation of the package.

We use *fontinst* by preparing a simple `plain` TEX file. Typically, this file will be short, and will consist of a command to `\input` the `fontinst.sty`, followed by a variety of commands which tell *fontinst* how to create the virtual font. Normally, `vf` and `tfm` files are binary files, file types which TEX cannot write. Therefore, *fontinst* reads and writes property list files and special metric and encoding files instead. These are all in ASCII, and the property files in particular are ASCII equivalents to `vf` and `tfm` files with extensions `vpl` and `pl`. Part of your TEX installation should include the utilities `vptovf` and `pltotf` (together with their inverses `vftovp` and `tftopl`), and we would then use these utilities to create the font files we need.

After each successful run of *fontinst* there will be three new kinds of files in your working directory.

- `pl` files — one for each raw font — which feeds into `pltotf` to create a `tfm` file;
- `vpl` files — one for each virtual font — which feeds into `vptovf` to create one `vf` and one `tfm` file; and
- an `fd` *font descriptor file* for each font family which NFSS will use to relate the font attributes to individual fonts.

(There are also some new `mtx` files and the usual `log` file that you can delete.) It is necessary to run all `vpl` files through `vptovf` and all `pl` files through `pltotf` to generate the binary metric files that TEX needs. A map file, such as `psfonts.map` for *dvips*, must be updated.

All `tfm` files belong with your other `tfm` files. The `vf` files belong in a special place as well, where *dvips* expects to find virtual files. The `fd` files belong in a TEX inputs directory.

6.1 Installing fontinst

The *fontinst* package consists of `fontinst.sty`, additional files, some documentation, some samples, and many examples. You may well receive the package as a zipped collection of files already organized in its own directory structure. I found it convenient to create a `.../fontinst` directory in which I unpacked *fontinst*. One or two levels down is a new directory called `inputs`. In addition to `fontinst.sty` itself, there is a collection of files with extensions `mtx` and `etx`. Move these files to one of your TEX input directories to complete the installation.

The *fontinst* package provides a new language for the creation of virtual fonts of all types. Our goal shall be to develop familiarity with these procedures so we can install any font with (relatively) little work.

7 Simple font installation with fontinst

7.1 New commands

Figure 3 displays one way to use *fontinst* to install the Times Roman fonts that are resident in every PostScript printer. Most *fontinst* installation files resemble this display.

Much of this file is standard boilerplate. The first line

```
\input fontinst.sty
```

makes *fontinst* known to TEX.

The pair of commands `\installfonts` and `\endinstallfonts` (with no arguments) surrounds the sequence of commands that do the bulk of the work. One or more `\installfamily` commands now follow. The first argument specifies the encoding, the second the family designation, and the third a set of commands that will be executed each time the family is loaded. See the *fontinst* documentation for further details on this third argument; it will be empty in nearly all our work.

```
\input fontinst.sty

\installfonts
   \installfamily{OT1}{ptm}{}
   \installfont{ptmr7t}{ptmr8a,latin}%
      {OT1}{OT1}{ptm}{m}{n}{}
   \installfont{ptmrc7t}{ptmr8a,latin}%
      {OT1c}{OT1}{ptm}{m}{sc}{}
   \installfont{ptmri7t}{ptmri8a,latin}%
      {OT1}{OT1}{ptm}{m}{it}{}
   \installfont{ptmb7t}{ptmb8a,latin}%
      {OT1}{OT1}{ptm}{bx}{n}{}
   \installfont{ptmbc7t}{ptmb8a,latin}%
      {OT1c}{OT1}{ptm}{bx}{sc}{}
   \installfont{ptmbi7t}{ptmbi8a,latin}%
      {OT1}{OT1}{ptm}{bx}{it}{}
\endinstallfonts
\bye
```

Figure 3: One way to install Times Roman. This examples uses the original TEX encoding but does not include any expert fonts. Two series are installed — regular and bold. Within each series, three shapes are installed — upright, small caps (which use encoding file `OT1c.etx`), and italic.

```
\installfamily{encoding}{family}%
        {fd-commands}
```

The workhorse command in any installation file is the `\installfont` command, which takes eight parameters. The last parameter allows us to specify size information for the font. For scalable fonts, it is nearly always empty because scalable fonts are, well, scalable to any size. (Bitmap fonts, created specifically for different sizes, require non-empty entries.) Parameters 4 through 7 provide space for the encoding, family, series, and shape values that *fontinst* uses to create the NFSS `fd` file. The very first parameter stores the file name of the virtual font you want to create.

That leaves the second and third parameters. In order to understand their significance, we need a small digression to consider the process of font creation.

7.2 Creating fonts

There are two aspects to font creation:

1. **Metric:** We need procedures for constructing each glyph or character in the virtual font. I use

the term 'metric' to encompass these structural aspects, which ultimately involve measuring the glyphs for use by TEX.

2. **Encoding:** We need to decide on the order of the glyphs in the font, and specify any additional rules that the characters need to live by. For example, rules might concern ligatures (any time an i follows a single f, replace it by fi; any time an *A* appears at the beginning of a word, replace it by a swash variant), or math symbols (any time interior material gets too tall, replace a delimiter by the next larger size).

For *fontinst*, these instructions should be in *metric files*, with an `mtx` extension, and *encoding files*, with extension `etx`. In the second position of the `\installfont` command, we place a list of metric files to be inserted. *fontinst* reads them to find out how to construct the characters. The third position records the name of an encoding file, which *fontinst* reads to learn which characters to include, how to order them, and what ligature and other special rules to follow.

An `\installfont` instruction looks like this.

```
\installfont{font-name}{metric-files}%
      {encoding-file}{encoding}%
      {family-name}{series}{shape}{size}
```

7.3 Metric files

The task of preparing metric files is lightened because *fontinst* reads three types of metric files:

1. `mtx` files, using a format specific to *fontinst*;
2. `afm` files, the ASCII metric files that come with each scalable outline font; and
3. `pl` files, the ASCII equivalents to a TEX `tfm` file.

fontinst reads the first two types automatically, but you will need to use the program `tftopl` (which should accompany your version of TEX) to create the third file. For example, type

```
tftopl cmr10.tfm cmr10.pl
```

to do the obvious thing.

In *fontinst* prior definitions take precedence over subsequent definitions. That is, if any construct appears more than once in a series of files that *fontinst* reads, only the first one counts; later definitions are silently ignored. Therefore, *the order in which fontinst reads files is critical!* This philosophy is central to the way *fontinst* works, as we'll see.

The file `latin.mtx` is the "metric file of last resort." It provides instructions for creating 401 glyphs found in Latin alphabets. Of those 401, some are unfakable — there's no way to print a character like 'A' unless the A is in the font, but many other glyphs can be faked. Accented letters can be built from letters and accents, and small caps can be taken from an uppercase font set at 80% of the current design size. Of course, there isn't room for all 401 of these characters in a single font anyway. (The limit is 256.) But because many of these characters have been previously defined in metric files, *fontinst* will ignore many of the definitions in `latin.mtx` — remember, glyph constructs have no effect if defined previously. But if you have neglected to define a glyph that you later call for, the definitions in `latin.mtx` serve as safety net. That is why all the `\installfont` commands in figure 3 and in virtually every *fontinst* example contain lists of metric files that terminate with a call to `latin.mtx`.

7.4 Encoding files

Once the metric files have done their job (of constructing the glyphs), a single encoding file chooses the group of characters that belong in the font and the proper order (encoding). This file also specifies certain ligature and other rules for the font to abide by.

Encoding files tend to have names that reflect their encoding. Thus, the encoding file for the OT1 encoding is simply `OT1.etx`. Similar files, `OT1c.etx` and `OT19.etx`, would set up a small caps and an old-style figures font using OT1 encoding. There are several more variants in the *fontinst* distribution.

8 Progressive examples

It's time to consider examples using *fontinst* to create virtual fonts.

8.1 Simple font installation

The simplest way to use *fontinst* is to run TEX on the file `fontinst.sty` and to then type

```
\latinfamily{ptm}{}
\bye
```

in response to TEX's star prompt `*`. This works presuming that all the fonts in the `ptm` family (Times Roman) have been named in accordance with Karl Berry's font naming scheme and that all font metric files are in places that TEX can read from.

This method is best for authors who plan never to need any more exotic fonts than these. Subsequent examples are designed to show off the power of *fontinst* and to teach its intricacies in a tutorial manner.

8.2 Easy EC fonts

The Cork encoding, denoted by T1, refers to the standard agreed upon at a TEX meeting held in Cork, Ireland, in September 1990. There, agreement was reached for sets of 256-character fonts for use by TEX. (The TEX standard had at that time only been extended to 256 character fonts for a short time.) The `ecr` fonts look like the usual Computer Modern fonts, but these fonts have been extended according to the Cork standard. Virtual fonts provide an easy way to generate `ecr` fonts from raw, Computer Modern fonts.

For each virtual `ecr` font, a corresponding `cmr` font acts as the single raw font. We will need the property list `pl` file as well.

Here are the steps to create a virtual `ecr10` from a raw `cmr10` font. The installation file `makeecr.tex` should resemble

```
% This is makeecr.tex.
\input fontinst.sty
  \installfonts
    \installfamily{T1}{ecr}{}
    \installfont{ecr10}{cmr10,latin}%
        {T1}{T1}{ecr}{m}{n}{}
  \endinstallfonts
\bye
```

although you'll need additional `\installfont` statements for members of this family which are italic, boldface, and so on.

After running this installation file through TEX, enter these statements at the prompt:

```
tftopl cmr10.tfm cmr10.pl
tex makeecr
vptovf ecr10.vpl ecr10.vf ecr10.tfm
rm *.log *.pl *.vpl *.mtx
```

after which you'll need to move the `tfm` and `vf` files to their proper places. In words, we need first the ASCII property list file, after which we can invoke TEX and *fontinst*. Thereafter, we create binary font files using the virtual property `vpl` produced by *fontinst*. Finally, we clean up. (Unix syntax is shown.) This example does not require an addendum to `psfonts.map` unless you are using scalable versions of the Computer Modern fonts.

8.2.1 Easy ecr10: additional comments

During the creation of `ecr10.vpl`, *fontinst* reports 34 glyphs are missing—that is, of the full complement of characters that do belong in a T1-encoded font, *fontinst* complained 34 times that it couldn't make the glyph. All of these are various diacritics (ring, ASCII tilde, and so on) and accented letters that use these missing diacritics, but a few are more problematic, including the sterling symbol and French quotations. If you access these characters, the mock `ecr10` font will not be suitable.

I have found there is no premium on disc space from using these fonts. The `vf` and `tfm` files require roughly 4k and 5.7k apiece, comparable with an actual `pk` file at a laser printer resolution.

8.3 Installing outline fonts

The *vfinst* utility takes care of scalable font installation, but we are now in a position to understand a simple installation ourselves. We may begin by renaming the font files to conform to a TEX font naming standard. Suppose we have Adobe Garamond Roman fonts to install. We rename the regular font files to `padr8a.pfb` and `padr8a.afm`, for example.

As an example, we can create the OT1-encoded font `padr7t` from these. This new font will belong to font family `pad` and have NFSS designations of `m` and `n` (medium series, normal shape). With this information, we prepare an installation file that looks like

```
% This is file makepad.tex
\input fontinst.sty

\installfonts
 \installfamily{OT1}{pad}{}
 \installfont{padr7t}{padr8a,latin}%
    {OT1}{OT1}{pad}{m}{n}{}
\endinstallfonts
```

although a real installation will likely contain several `\installfont` commands. The `\installfont` command is quite straightforward. It:

- constructs a font for family `pad`;
- uses glyph information from `padr8a.afm`, and supplements it (if necessary) with instructions from `latin.mtx`;
- applies the OT1 encoding to it; and
- uses the four parameters `OT1`, `pad`, `m`, and `n` for the NFSS `fd` file.

8.3.1 Incorporating expert fonts

For the vast majority of outline fonts, the only way to get the ff, ffi, and ffl ligatures is from an expert font, because these characters are rarely present in a base font. However, `latin.mtx` does create mock characters with these names because slots are provided for these ligatures in the font by the encoding files. Therefore, the way to get the honest double-f ligatures is simply to include the expert font name in the list of metric files in an `\installfont` command. That is, the skeletal installation file listed above would look something like

```
% This is file makepad.tex
\input fontinst.sty

\installfonts
\installfamily{OT1}{pad}{}
\installfont{padr7t}{padr8a,padr8x,latin}%
  {OT1}{OT1}{pad}{m}{n}{}
\endinstallfonts
```

Note the presence of `padr8x`, the expert font for Garamond regular. (This discussion is presented for pedagogical completeness only, for in this case it's better to use `\latinfamily` (see above, section 8.1) or other font installation tools that do much of this grunt work for you.)

8.4 Conclusion

These examples only just begin to demonstrate the power and utility of virtual fonts. Other of my own virtual font projects involve letterspacing (tracking), new families of math fonts, underline and strike-out fonts, oldstyle figure fonts, better footnote numbers from expert fonts, and roman families that incorporate swash or alternate characters if those characters are present. I use virtual techniques to add dotless j's to my roman fonts even when they don't exist in the original raw font. As far as I can tell, the TEX community still has a great deal to do to educate itself properly in the use and construction of virtual fonts. I hope my remarks in this brief article offer preliminary steps towards this goal.

◇ Alan Hoenig
17 Bay Avenue
Huntington, NY 11743
ajhjj@cunyvm.cuny.edu

Abstracts

Les Cahiers GUTenberg
Contents of Issue 25

Numéro 25 — novembre 1996

Editor's note: While this is the first non-thematic issue since no. 18 (!), there still seems to be a certain cohesiveness running through four of the six articles: issues and uses of TEX in real situations. They are as interesting for their examination of the respective problems as for the history of change they reflect — change in fonts, change in TEX installations, change in language. As a side note, this issue has no lead editorial to introduce the articles.

THIERRY BOUCHE, Sur la diversité des fontes mathématiques [Regarding the diversity of math fonts]; pp. 1–24

We are interested in the issues which arise when changing the fonts used by LATEX's math modes. We will try to provide some concrete solutions for accessing a larger variety of fonts, without falling victim to typographic nonsense.

[from the Résumé]

[The author's abstract belies the significance and interest of his article, which explores in detail the issues which surround this particular problem with TEX math material: the need for a greater variety of font combinations for text and maths. Starting with CM, Lucida, and MathTime, the author then experiments with `mathptm` and `mathfont` (his own virtual font), in combination with the text fonts Apollo and Utopia. Several examples of text + math are shown,[1] along with details of the LATEX 2ε coding.]

CHRISTIAN LENNE, Édition structurée et non structurée d'expressions mathématiques dans Thot [Structured and non-structured editing of math expressions, using Thot]; pp. 25–32

Document writing in the scientific world or leading-edge industries must have the tools/options for expressing mathematical formulas. Commercial word processing tools offer little such functionality, or can offer only that functionality. In this article, we will present the approach we have used with the Thot structured document editor.

[from the Résumé]

[1] It is my great misfortune to have a copy with several faintly printed pages.

Jean-Pierre Vial, Utilisation de Y&YTEX en langue française [Using Y&YTEX in French]; pp. 33–40

This article discusses the experiences of a "normal" TEX user — as opposed to a TEXpert, that is — in "Frenchifying" and using Y&YTEX.

[from the Introduction]

[The article begins with a brief overview of Y&YTEX, followed by some comments on why the implementation is of particular interest to the European TEX user: the ease of use and access to PS fonts not only for printing but also previewing, and recoding options, particularly of accented characters. Incompatibility of encodings arises when source files meet TEX and when `dvi` output files meet previewers. Y&Y solutions on both fronts are discussed, with particular reference to the three main components of the package: YandYTEX, the newly renamed TEX implementation; DVIWINDO, the previewer; and DVIPSONE, the printer driver. The article concludes with a run-down of the font families available in PS format: the CM fonts, LucidaBright (including the expert set), and mathtime, a maths font to go with Adobe's Times Roman. The author says: "Other than the systematic approach to PS, there's nothing revolutionary here ... What is worth noting is the coherence of the whole system ... the communication across processes."]

Marc Torzynski, Histoire de TEX sous Dos et Windows à l'École nationale supérieure de physique de Strasbourg [TEX under Dos and Windows at the National College of Physics in Strasbourg]; pp. 41–56

By outlining the evolution of the TEX installation at his site, the author shows how he came to develop various working environments under DOS and Windows. The tools have been installed on the facility's intranet and are available to both staff and students.

[from the Résumé]

[Quite a detailed history, beginning with "Life before TEX", "The Arrival of TEX at the College", and "Discovering TEX for the PC". Then begins the real work: installing TEX on an MS-DOS network, and the eventual migration to Windows for Workgroups. After a year and a half, the author feels he can say: "The basic problems seem to have been identified and corrected, and the working environment is now stable."]

Bernard Gaulle, Quelques questions de droit français à propos des logiciels sur Internet [Some issues regarding French law and programs on the Internet]; pp. 57–64

[The author discusses various legal issues on such topics as author's rights, user licences, auditing, public domain, the law on the Internet, test and demonstration programs, shareware, freeware, the implications for CDs, discs, and other media, and so on.]

Jacques André, ISO Latin-1, norme de codage des caractères européens ? trois caractères français en sont absents ! [ISO Latin-1, encoding standard for European characters? Three French characters are missing!]; pp. 65–77

Three French characters (œ, Œ, and Ÿ) are missing in ISO Latin-1. It is explained why they should be there and why they are not. Then, some comments are made on the standardisation of characters.

[from the author's abstract]

[As I have come to expect from Jacques, when he starts talking about characters, a great deal of information provided within an extremely knowledgeable historical context, backed up with wonderful bibliographic citations. Takes me back to my courses in historical linguistics!]

[Compiled by Christina Thiele]

Articles from *Cahiers* issues can be found in PostScript format at the following site:
`http://www.univ-rennes1.fr/pub/GUTenberg/publicationsPS`

Die TEXnische Komödie
Contents of Recent Issues

7. Jahrgang, Heft 1/1995 (Mai 1995)

Luzia DIETSCHE, [Editorial]; p. 3
A short statement commenting on the current issue, which 'contains more technical than TEXnical material'.

◦ *Hinter der Bühne : Vereinsinternes*
[Backstage : Club matters]; pp. 4–20:

Joachim LAMMARSCH, Grußwort
[Welcome message]; pp. 4–5
A short comment on club matters by the president of DANTE. In particular, he mentions that after his resignation as a special director of TUG all other special directors also left TUG's Board of Directors. Consequently local groups are no longer represented on it.

[Luzia DIETSCHE],
Protokoll der 12. Mitgliederversammlung von

DANTE, Deutschsprachige Anwendervereinigung TEX e.V. [Protocol of the 12th assembly of members of DANTE]; pp. 5–18

This is the official report on the members' meeting held in Gießen (March 2, 1995). It starts with the announcement that a member has been excluded from DANTE by the presidium, then it continues (as usual) with short accounts on the various hardware platforms, `german.sty` and other special topics (as presented by the appointed coordinators), followed by a report on the situation of DANTE (mainly organizational matters), and its activities (including among others the plans for a CD containing parts of the CTAN archive, the installation of a mailbox, and the status of ε-TEX). The results of the election of the board members are given.

Jürgen UNGER, Die Mailbox von DANTE e.V. – es ist vollbracht [The DANTE e.V. mailbox – done!]; pp. 18–20

DANTE has installed a mailbox for access by modem.

◦ *TEX-Theatertage*
[TEX theatre festival]; pp. 21–25:

Thomas FEUERSTACK, Der Zauber eines Lächelns [The magic of a smile]; pp. 21–25

A lively and personal account of DANTE '95 (Feb. 28–March 3, 1995). One of the highlights was an exhibition of the calligraphies from Knuth's *'3:16 – Bible Texts Illuminated'*.

◦ *Von fremden Bühnen*
[On other stages]; pp. 26–27:

Frank MITTELBACH [for the LATEX3 Project], Math Font Encoding (in English); pp. 26–27

An announcement of the final report on 'Math Font Encoding' (`13d007.tex` and `13ms002.cls` in `/tex-archive/info/ltx3pub`, 90 pages) by Justin Ziegler.

◦ *Bretter, die die Welt bedeuten*
[The stage is the world]; pp. 28–30:

Siegfried SPLETT, Das russische Alphabet mit "Bordmitteln" erstellt [The Russian alphabet – done "on board"]; pp. 28–30

A short note on producing Russian letters in the `picture` environment of LATEX.

◦ *TEX-Beiprogramm* [TEX co-features]; pp. 31–42:

Markus PORTO, Weiterhin guten Appetit! [May it continue to taste good!]; pp. 31–32

The interactive LATEX cookbook installed at the WWW-server of Gießen (issue 2/1994, p. 43) is now available as a stand-alone HTML-version.

Bericht des technischen Beirats [Report of the technical council]; pp. 32–42

A collection of short reports (by the appointed coordinators) on the various hardware platforms and other special topics, supplementing those of the protocol (p. 5).

◦ *Rezensionen* [Reviews]; pp. 43–44:

Luzia DIETSCHE, TEX: starting from 1; pp. 43–44

A review of Michael Doob's introduction to plain TEX. The bottom line: '...for all who want to get a first impression'.

◦ *Leserbrief(e)* [Letter(s)]; pp. 45–50:

Jan BRAUN, WinWord versus LATEX, Heft 4/1994 [WinWord versus LATEX, issue 4/1994]; pp. 45–48

A letter in defense of LATEX, arguing that the 'intuitive' guidance offered by WinWord and similar WYSIWYG programs is only superficially more user-friendly than LATEX with, for example, an `emacs`-shell. (LA)TEX makes the experience of four centuries of typography available, and is one of the few possibilities — if not the only one — to get high-quality output: 'What you get is what you *want*!'

Georg BAUHAUS, Gedanken zu Gedanken zu Gedankenstrichen [Thoughts on thoughts on dashes]; pp. 48–50

In his answer to an article by Martin Schröder, the author pleads for more tolerance of (intentional) deviations from the typographical tradition as presented by the DUDEN (the 'bible' of correct German), e.g., where the 'correct' form of a dash (which differs from American usage) is concerned.

◦ *Spielplan* [Repertory]; pp. 51–59:

The international and national calendar, and announcements of conferences (16th Annual TUG Meeting 'Real World TEX' and EuroTEX 1995 'The TEX Toolbox').

◦ *Adressen* [Addresses]; pp. 60–63:

Various addresses related to DANTE, the addresses of everyone who has contributed to this issue, and the addresses of the coordinators in

charge of the various hardware platforms and other special topics.

7. Jahrgang, Heft 2/1995 (September 1995)

Luzia DIETSCHE, [Editorial]; p. 3
This issue is the first one prepared with a set of new macros which (also) can be used with LaTeX2ε.

◦ *Hinter der Bühne : Vereinsinternes* [Backstage : Club matters]; pp. 4–5:

Joachim LAMMARSCH, Grußwort [Welcome message]; pp. 4–5
A short comment on club matters by the president of DANTE: Jürgen Unger (responsible for the mailbox) has been appointed member of the technical council. There have been problems concerning TUG membership dues, and Luzia Dietsche also has left the TUG Board of Directors. Consequently, DANTE is no longer represented in it.

◦ *Von fremden Bühnen* [On other stages]; pp. 6–10:

LaTeX3 PROJECT TEAM, Modifying LaTeX2ε (in English); pp. 6–10
'This document describes the principles underlying our policy on distribution and modification of the files comprising the LaTeX system.'

◦ *Bretter, die die Welt bedeuten* [The stage is the world]; pp. 11–35:

Gerd NEUGEBAUER, Setzen russischer Textteile mit LaTeX [Typesetting portions of Russian text in LaTeX]; pp. 11–20
This article shows how text in Cyrillic fonts can be inserted into normal text. In particular, the Cyrillic fonts from the University of Washington (`wncyr`, etc.) are described.

Bernd RAICHLE, Orale Spielereien mit TeX – Teil II [Oral games with TeX – part II]; pp. 20–30
After a description of the components of TeX, the program, and the analogy to the *'anatomy of TeX' —mouth, eyes, gullet, stomach, bowels—* expandable (and non-expandable) tokens and some (tricky) examples concerning numbers are discussed in detail. (The solution to a question posed in part I is given.)

Matthias MALEK, Style-Files – leicht gemacht – zum Zweiten [Style files made easy, second time around]; pp. 31–35
The author presents LaTeX macros which allow the typesetting of labels (for envelopes), improving those described earlier by Arne W. Steuer (issue 6/3, 1994)

◦ *Was Sie schon immer über TeX wissen wollten ...* [What you always wanted to know about TeX...]; pp. 36–37:

Bernd RAICHLE, Datumsangaben [Inserting the date]; pp. 36–37
Some hints on inserting a customized version of the current date.

◦ *TeX-Beiprogramm* [TeX co-features]; pp. 38–45:

Jürgen HANNEDER, TeX unter OS/2 für akademische Minderheiten [TeX under OS/2 for academic minorities]; pp. 30–41
Report on installing emTeX, AucTeX under OS/2 and adapting the installation for convenient use of the `wsuipa` (phonetic) fonts.

Gerd NEUGEBAUER, Kurioses aus dem Fundus [Curiosities from the property room]; pp. 41–43
The author points out that font changes in LaTeX have unexpected side effects: Some sequences (e.g. `\small\large` and `\normalsize\large`) may lead to different baselineskips.

Frankeye JONES, The Underfull Badness Blues (in English); p. 44
A song to the tune of "Do Run Run".

[Kleinanzeige] [Advertisement]; p. 45
A computer specialist is looking for a job.

◦ *Rezensionen* [Reviews]; pp. 46–47:

Luzia DIETSCHE, Fachwörterbuch Kommunikationsdesign [Dictionary of communication design]; pp. 46–47
A review of *'Fachwörterbuch Kommunikationsdesign'* by Petra Wilhelm (Springer 1995), a reference on typography, design and graphics.

◦ *Spielplan* [Repertory]; pp. 48–51:
The international and national calendar, and the announcement of a conference (DANTE'96 in Augsburg).

◦ *Adressen* [Addresses]; pp. 52–55:
Various addresses related to DANTE, the addresses of everyone who has contributed to this issue, and the addresses of the coordinators in charge of the various hardware platforms and other special topics.

7. Jahrgang, Heft 3/1995 (Dezember 1995)

Luzia DIETSCHE, [Editorial]; p. 3

A short overview of the contents of the issue—a lot of technical and TEXnical material.

◦ *Hinter der Bühne : Vereinsinternes* [Backstage : Club matters]; pp. 4–21:

Joachim LAMMARSCH, Grußwort [Welcome message]; pp. 4–5

A short comment on club matters by the president of DANTE. Among others he mentions the production of the first CD-ROM in cooperation with Addison-Wesley (a supplement to a book that will be available separately for members), and the final outcome of the exclusion of a member (he has withdrawn his accusations).

[Luzia DIETSCHE], Protokoll der 13. Mitgliederversammlung von DANTE, Deutschsprachige Anwendervereinigung TEX e.V. [Protocol of the 13th assembly of members of DANTE]; pp. 6–20

This is the official report on the members' meeting held in Berlin (September 14, 1995). After discussing and confirming the exclusion of a member who had accused DANTE officials of manipulating an election, the assembly continues (as usual) with short accounts on the various hardware platforms (as presented by the appointed coordinators), and reports on other special topics such as TUG, TUG'95, EuroTEX'95.

Joachim LAMMARSCH, Die Ära des Präsidenten geht zu Ende [The era of the president ends]; pp. 20–21

The president announces his (forthcoming) resignation and looks back on ten years of work for DANTE. He thinks that Luzia Dietsche would be a suitable successor.

Joachim LAMMARSCH, Fonds zur Unterstützung von Mitgliedern [Funds to subsidize members]; p. 21

A short announcement that (under certain conditions) members can apply for a reduction in membership dues.

◦ *Bretter, die die Welt bedeuten* [The stage is the world]; pp. 22–33:

Bernd RAICHLE, TEX Capacity exceeded ... – Teil I [TEX Capacity exceeded ... – part I]; pp. 22–33

The author discusses the various errors which lead to the error message cited in the title of this article. In this first part the topics treated in detail are `buffer size` (long lines), `input stack size`, and `text input levels`.

◦ *TEX-Beiprogramm* [TEX co-features]; pp. 34–51:

Bericht des technischen Beirats [Report of the technical council]; pp. 34–43

A collection of short reports (by the appointed coordinators) on the various hardware platforms and other special topics, supplementing those of the protocol (p. 5). Among others, a new German book is mentioned: *LATEX Tips* by Kenneth Shultis (a translation of the 1994 book on LATEX 2.09).

Joachim SCHROD, TDS – Die vorgeschlagene TEX Directory Structure [TDS – A proposal for a TEX directory structure]; pp. 44–47

This article reports on 15 months of work done by the TUG committee and describes the aims of a TDS standard, but does not describe the proposal.

Joachim LAMMARSCH, Beschreibung der CD-ROM von DANTE e.V. [Description of the DANTE CD-ROM]; pp. 48–50

A short description of the CD-ROM produced by DANTE (cf. p. 4). It is an excerpt from a book (*CTAN/3 – Das TEX/LATEX-Archiv von DANTE e.V.*).

Norbert HESSE, Erfinder von TEX, LATEX und METAFONT geben zu: Alles Betrug!!! [The inventors of TEX, LATEX and METAFONT admit: all fraud!!!]; pp. 50–51

A (satirical) article inspired by a text of Bernhard L. Hayes: Lamport and Knuth admit that their work was meant as a parody.

◦ *Rezensionen* [Reviews]; pp. 52–53:

Gerd NEUGEBAUER, Wie verfaßt man wissenschaftliche Arbeiten? [How to write a scientific paper]; pp. 52–53

This is a review of: Klaus Poenicke, *Duden. Wie verfaßt man wissenschaftliche Arbeiten?* Though this book, originally published in 1988, seems to contain a reference to TEX, it does not sufficiently treat the rôle of computers. (DUDEN is a major publisher of dictionaries, lexicons and handbooks.)

◦ *Spielplan* [Repertory]; pp. 54–55:

The international and national calendar.

◦ *Adressen* [Addresses]; pp. 56–59:

Various addresses related to DANTE, the addresses of everyone who has contributed to this

issue, and the addresses of the coordinators in charge of the various hardware platforms and other special topics.

7. Jahrgang, Heft 4/1995 (März 1996)

Luzia DIETSCHE, [Editorial]; p. 3

The author announces a special LaTeX2ε class (cf. p. 6) for the *Komödie*, which soon will get a new layout design. Moreover, she gives a short overview of the contents of the current issue.

◦ *Hinter der Bühne : Vereinsinternes* [Backstage : Club matters]; pp. 4–5:

Joachim LAMMARSCH, Grußwort [Welcome message]; pp. 4–5

A short comment on club matters by the president of DANTE. Among other topics he discusses various CD-ROMs which contain TeX material.

◦ *Bretter, die die Welt bedeuten* [The stage is the world]; pp. 6–40:

Gerd NEUGEBAUER, Eine Klasse für die "Die TeXnische Komödie" [A class for "Die TeXnische Komödie"]; pp. 6–15

The author describes the LaTeX2ε document class used for production of the *Komödie* (mainly developed by himself).

Bernd RAICHLE, Orale Spielereien mit TeX – Teil III [Oral games with TeX – part III]; pp. 15–29

The third instalment of the series discusses situations where TeX inserts a `\relax` token which, in some cases, may be annoying (e.g., in a `\message`), but which also can be used to devise 'dirty tricks' (as explained using a tricky conditional statement credited to David Kastrup). (The answers to two questions posed in part II are given.)

Werner LEMBERG, Das CJK-Paket für LaTeX2ε [The CJK package for LaTeX2ε]; pp. 30–40

The CJK package (Chinese/Japanese/Korean) supports the simultaneous use of different multibyte coding schemes under LaTeX2ε. It is designed to work with the MULE (multilingual Emacs) editor.

◦ *Was Sie schon immer über TeX wissen wollten ...* [What you always wanted to know about TeX ...]; pp. 41–42:

Bernd RAICHLE, Kapitälchen in Überschriften [Caps and small caps in headings]; pp. 41–42

The author points out that the standard fonts do not provide a bold caps-and-small-caps font and that, as a result, LaTeX2ε uses bold roman font when logos defined using `\textsc` appear in a heading. He shows how to use `\DeclareFontShape` to get (normal) caps and small caps instead.

◦ *TeX-Beiprogramm* [TeX co-features]; pp. 43–47:

Andreas DAFFERNER, Luzia DIETSCHE, Bernd RAICHLE, Volker RW SCHAA, and Rainer SCHÖPF, Das alternative (LA)TeX-Glossar [The alternative (LA)TeX glossary]; pp. 43–47

This is a collection of (short) humorous definitions of LaTeX commands and terms.

◦ *Rezensionen* [Reviews]; pp. 48–52:

Rainer HÜLSE, 4allTeX – eine Ready-to-Run-CD auf dem Prüfstand [4allTeX – testing a ready-to-run CD]; pp. 48–52

The author describes his experiences with the 4allTeX distribution (3rd edition) by the NTG (the TeX users group of the Netherlands) including a look back on the 2nd edition. His overall impression is good, but he makes some constructive critical remarks. His bottom line: The main flaw of the installation is that it assumes the use of 4DOS.

◦ *Spielplan* [Repertory]; pp. 53–59:

The international and national calendar, and the announcement of TUG'96 (Poly-TeX) in Dubna.

◦ *Adressen* [Addresses]; pp. 60–63:

Various addresses related to DANTE, the addresses of everyone who has contributed to this issue, and the addresses of the coordinators in charge of the various hardware platforms and other special topics.

(compiled by Peter Schmitt)

Late Breaking News

Production Notes

Mimi Burbank

Production of this issue entailed considerable "testing" of software on quite a few different platforms located here at the Supercomputer Computations Research Institute (SCRI) at Florida State University — in preparation for the release of the **TEX Live** CD-ROM.

The CDs were prepared in the United Kingdom, and shipped to Cadmus Journal Services in Easton, Maryland for inclusion in this issue. This CD-ROM distribution is a joint effort by the TEX Users Group, the UK TEX Users Group, and the French TEX Users (GUTenberg), with the support of the Dutch, German and Czech/Slovak user groups.

Output The final camera copy was prepared at SCRI on an Alphastation 2100/500 running OSF1 v3.2, using the *TeX Live* setup (Version 2), which is based on Karl Berry's *Web2c* TEX implementation version 7.0. PostScript output at 600dpi was produced using Radical Eye Software's dvipsk 5.66a and printed on a QMS 860 printer.

Coming Next Issue

The next issue of *TUGboat* will be the acroTUG'97 Proceedings issue. For more information on the topical information, please refer to the TUG'97 provisional program on page 139. The next regular issue will be the December issue, in which we plan to offer you an article by B. Jackowski on "METAFONT: practical and impractical applications", and a proposal by Pedro Aphalo entitled "A proposal for citation commands in LATEX3", as well as more material for new users. We also plan to provide a list of commonly-used acronyms using throughout the year in the various issues of *TUGboat*.

◇ Mimi Burbank
SCRI, Florida State University,
Tallahassee, FL 32306 – 4052
mimi@scri.fsu.edu

LaTeX

Modifying LaTeX*

LaTeX3 Project Team

Abstract

This document was produced in response to suggestions that the modification and distribution conditions for the files constituting the New Standard LaTeX system should be similar to those implied by Version 2 of the GNU General Public Licence, as published by the Free Software Foundation.

Introduction

This article describes the principles underlying our policy on distribution and modification of the files comprising the LaTeX system. It has been produced as a result of detailed discussions of the issues involved in the support and maintenance of a widely distributed document processing system used by diverse people for many applications. These discussions have involved users, maintainers of installations that support LaTeX and various types of organisations that distribute it. The discussions are continuing and we hope that the ideas in this article will make a useful contribution to the debate.

Our aim is that LaTeX should be a system which can be trusted by users of all types to fulfill their needs. Such a system must be stable and well-maintained. This implies that it must be reasonably easy to maintain (otherwise it will simply not get maintained at all). So here is a summary of our basic philosophy:

> We believe that the freedom to rely on a widely-used standard for document interchange and formatting is as important as the freedom to experiment with the contents of files.
>
> We are therefore adopting a policy similar to that which Donald Knuth applies to modifications of the underlying TeX system: that certain files, together with their names, are part of the system and therefore the contents of these files should not be changed unless the following conditions are met:
>
> - they are clearly marked as being no longer part of the standard system;
> - the name of the file is changed.

The system

In developing this philosophy, and the consequent limitations on how modifications of the system should be carried out, we were heavily influenced by the following facts concerning the current widespread and wide-ranging uses of the LaTeX system.

1. LaTeX is not just a document processing system; it also defines a language for document exchange.
2. The standard document class files, and some other files, also define a particular formatting of a document.
3. The packages that we maintain define a particular document interface and, in some cases, particular formatting of parts of a document.
4. The interfaces between different parts of the LaTeX system are very complex and it is therefore very difficult to check that a change to one file does not affect the functionality of both that file and also other parts of the system not obviously connected to the file that has been changed.

   This leads us to the general principle that:

   > with certain special exceptions, if you change the contents of a file then the changed version should have a different file name.

We certainly do not wish to prevent people from experimenting with the code in different ways and adapting it to their purposes. However, we are concerned that any distribution of modifications to the code should be very clearly identified as not being a part of the standard distribution. The exact wording and form of the distribution conditions is thus something that is flexible, but only within the constraint of keeping LaTeX as a standardised, reliable product for the purposes described above: the exchange and formatting of documents.

Some examples

Here we elaborate the arguments that have led us to the above conclusion.

Separate development considered harmful!

In many fields, the use of LaTeX as a language for communication is just as important as its capacity for fine typesetting; this is a very important consideration for a large population of authors, journal editors, archivists, etc.

Related to this issue of portability is the fact that the file names are part of the end-user syntax.

As a real example, the LaTeX 'tools' collection contains the package 'array.sty'. A new user-level feature was added to this file at the end of 1994 and a document using this feature can contain the line:

```
\usepackage{array}[1994/10/16]
```

By supplying the optional argument, the document author is indicating that a version of the file `array.sty` dated no earlier than that date is required to run this document without error.

This feature would be totally worthless if we were to allow an alternative version of the array package to be distributed under the same name since it would mean that there would be in circulation files of a later date, but without the new feature. If the document were processed using this 'alternative array' then it would certainly produce 'undefined command' errors and would probably not be processable at all.

What's in a file-name?

In a pure markup language, such as SGML, it is reasonably clear that control over the final presentation lies with the receiver of a document and not with the author.

However, the way that LaTeX is often used in practice means that most people (at least when using the standard classes and packages) expect the formatting to be preserved when they send the document to another site.

For example, suppose, as is still the most common use of LaTeX in publishing, you produce a document for 'camera-ready-copy' using the class 'article' and that you carefully tune the formatting by, for example, adding some explicit line breaks etc, to ensure that it fits the 8 page limit set by the editor a journal or proceedings.

It then gets sent to the editor or a referee who, without anyone knowing, has a non-standard version of the class file 'article' and so it then runs to 9 pages. The consequence of this will, at the least, be a lot of wasted time whilst everyone involved works out what has gone wrong; it will probably also lead to everyone blaming each other for something which was in fact caused by a misguided distribution policy.

It should also be noted that, for most people, the version of the class file 'article' that gets used is decided by a site maintainer or the compilers of a CD-ROM distribution. To most users, the symbols `article` in:

```
\documentclass{article}
```

are just as much part of LaTeX's syntax as are the symbols `12pt` in:

```
\hspace{12pt}
```

Thus they should both define a standard formatting rather than sometimes producing 1 more page or a 5pt larger space.

Users rely on the fact that the command (or menu item) 'LaTeX' produces a completely standard LaTeX, including the fact that 'article' is the 'standard article'. They would not be at all happy if the person who installed and maintains LaTeX for them were allowed to customise 'article' every second day so as (in her or his opinion) to improve the layout; or because another user wanted to write a document in a different language or typeset one with different fonts.

TeX itself

We have modelled our policies on those of the TeX system since this has for some time now been widely acknowledged as a very stable and high quality typesetting system.

The distribution policy set up by Donald Knuth for TeX has the following features:

- There is a clearly specified method for changing parts of the software by the use of 'change files'.
- Although arbitrary changes are allowed, the resulting program can be called TeX only if its functionality is precisely the same as that of TeX (i.e. neither less nor more) in all important areas.
- There are many files in the system that cannot be changed at all (without changing the name): examples are the file `plain.tex` and the files associated with fonts, including the Metafont source files.

Maintaining complexity

Our experience of maintaining LaTeX has shown us just how complex are the interactions between different parts of the system.

We have therefore, with lots of help from the bug reports you send in, developed a large suite of test files which we run to check the effects of every change we make. A non-negligible percentage of these test runs give unexpected results and hence show up some unexpected dependency in the system.

Some assurances

We are certainly not attempting to stop people reformatting LaTeX documents in any way they wish. There are many ways of customising incoming documents to your personal style that do not involve changing the contents of LaTeX's standard files;

indeed, this freedom is one of the system's many advantages. The simplest way to achieve this is to replace

`\documentclass{article}` by
`\documentclass{myart}`

Nor do we wish to discourage the production of new packages improving on the functionality or implementation of those we distribute. All we ask is that, in the best interests of all LaTeX users, you give your superbly improved class or package file some other name.

Configuration possibilities

The standard LaTeX system format can be configured in several ways to suit the needs and resources of an installation. For example, the loading of fonts and font tables can be customised to match the font shapes, families and encodings normally used in text mode. Also, by producing the appropriate font definition files, the font tables themselves can be set up to take advantage of the available fonts and sizes. The loading of hyphenation patterns can be adjusted to cover the languages used; this has to be done as part of making the format since this is the only stage at which patterns can be loaded.

A complete list of these configuration possibilities can be found in the distributed guide *Configuration options for* LaTeX 2$_\varepsilon$ (`cfgguide.tex`). However, as it says there, the number of configuration possibilities is strictly limited; we hope that having read this far you will appreciate the reasons for this decision. One consequence of this is that there is no provision for a general purpose configuration file, or for adding extra code just before the `\dump` of the format file.

This was a deliberate decision and we hope that everyone (yes, that includes you!) will support its intent. Otherwise there will be a rapid return to the very situation, of several incompatible versions of LaTeX 2.09, that originally prompted us to produce LaTeX 2$_\varepsilon$: the new, and *only*, 'Standard LaTeX'. This will make LaTeX unmaintainable and, hence, unmaintained (by us, at least).

> Therefore you should not misuse the configuration files or other parts of the distribution to produce non-standard versions of LaTeX.

Some of the allowed configurations can result in a system that can produce documents that are no longer 'formatting compatible'; for example, the use of different default fonts will most likely produce different line and page breaks. If you do produce a system that is configured in such a way that it is not 'formatting compatible' then you should consider carefully the needs of users who need to create portable documents. A good way to provide for their needs is to make available, in addition, a standard form of LaTeX without any 'formatting incompatible' customisations.

Modification conditions

It is possible that you need to produce a document processing system based on standard LaTeX but with functionality that cannot be implemented by using the approved configuration files and complying with the restriction on the code that is allowed in them. In other words, you may need a system which is sufficiently distinct from Standard LaTeX that it is not feasible to do this simply by using the configuration options we provide or by producing new classes and packages.

If you do produce such a system then, for the reasons described above, you should ensure that your system is clearly distinguished from Standard LaTeX in every possible way, including the following.

1. Give your system a distinguished name, such as NS-TeX, which clearly distinguishes it from LaTeX.
2. Ensure that it contains no file with a name the same as that of a file in the standard distribution but with different contents. (If this is not possible then you must:
   - ensure that files from the non-LaTeX system cannot be accidentally accessed whilst using a standard LaTeX;
   - ensure that each file from the non-LaTeX system clearly identifies itself as a non-LaTeX file on the terminal and in the log file.)
3. Ensure that the method used to run your system is clearly distinct from that used to run Standard LaTeX; e.g. by using a command name or menu entry that is clearly not `latex` (or `LaTeX` etc).
4. Ensure that, when a file is being processed by your system, the use of non-standard LaTeX is clearly proclaimed to the user by whatever means is appropriate.
5. Ensure that what is written at the beginning of the log file clearly shows that your system has been used, and that it is not Standard LaTeX. See the file `cfgguide.tex` for how to achieve this.
6. Clearly explain to users that bug reports concerning your system should not be sent to the maintainers of Standard LaTeX.

Note to system administrators If you install a non-standard (modified) version of LaTeX on a multi-user site then please, in addition, install Standard LaTeX and observe the conditions enumerated above, particularly 3.

What do you think?

We are interested in your views on the issues raised in this document. The best way to let us know what you think, and to discuss your ideas with others, is to join the `LaTeX-L` mailing list and send your comments there. To subscribe to this list, mail to:

`listserv@vm.urz.uni-heidelberg.de`

the following one line message:

`subscribe LATEX-L <firstname-surname>`

News from the LaTeX3 Project Team

LaTeX3 Project Team

Each release of LaTeX includes a short file containing a brief summary of the latest additions and enhancements.

These are carefully designed to fit a single page so that they can easily be displayed on notice boards, etc.

They all contain information that may be new to you, judging by the enquiries the team still get concerning information in the older issues.

The files are available on the Comprehensive TeX Archive Network CTAN: they are located in `tex-archive/macros/latex/base/` with names `ltnews##.tex`.

On the following seven pages, we publish, with thanks to the LaTeX3 team, the contents of LaTeX News issues 01 – 07, and we announce with pleasure that the team will be regularly providing information for the readers of *TUGboat*.

This material has not appeared in *TUGboat* so we include, unedited and in reverse order, all of the previous issues for those of you who might be interested in the work of the LaTeX3 project team and the evolution of LaTeX, as well as for those of you without network access and who are unable to "surf the net".

◇ LaTeX3 Project Team
Johannes Braams,
David Carlisle
Alan Jeffrey
Frank Mittelbach
Chris Rowley
and Rainer Schöpf

LaTeX News

Issue 7, June 1997

T1 encoded Computer Modern fonts

As in the last release the base LaTeX distribution contains three different sets of 'fd' files for T1 encoded fonts.

In this release the default installation uses `ec.ins` and so installs files suitable for the current 'EC fonts' distribution. If you have still not updated to the EC fonts and are using the earlier test versions, known as DC then you should unpack `newdc.ins` (for DC release 1.2 or later) or `olddc.ins` (for the original releases of the DC fonts). This should be done after unpacking `unpack.ins` but before making the format by running iniTeX on `latex.ltx`. There are further details in `install.txt`.

T1 encoded Concrete fonts

The Metafont sources for T1 encoded 'Concrete' fonts have been removed from the mfnss distribution as they were based on the now obsolete DC fonts release 1.1. Similarly the cmextra.ins install file in the base distribution no longer generates fd files for the 'Concrete' fonts. To use these fonts in either T1 or OT1 encoding it is recommended that you obtain Walter Schmidt's ccfonts package and fonts from CTAN `macros/latex/contrib/supported/ccfonts`.

Further input encodings

Two more inputenc packages have been added: for latin5, thanks to H. Turgut Uyar; and for latin3, thanks to Jörg Knappen.

Normalising spacing after punctuation

The command `\normalsfcodes` was introduced at the last patch release. This is normally given the correct definition automatically and so need not be explicitly set. It is used to correct a problem, reported by Donald Arseneau, that punctuation in page headers has always (in all known TeX formats) been potentially incorrect if the page break happens while a local setting of the space codes (for instance by the command `\frenchspacing`) is in effect. A common example of this happening in LaTeX is in the verbatim environment.

Accessing Bold Math Symbols

The tools distribution contains a new package, bm, which defines a command `\bm` that allows individual bold symbols to be accessed within a math expression (in contrast to `\boldmath` which makes whole math expressions default to bold fonts). It is more general than the existing amsbsy package; however, to ease the translation of documents between these two packages, bm makes `\boldsymbol` an alias for `\bm`.

This package was previously made available from the 'contrib' area of the CTAN archives, and as part of Y&Y's LaTeX support for the MathTime fonts.

Policy on standard classes

Many of the problem reports we receive concerning the standard classes are not concerned with bugs but are suggesting, more or less politely, that the design decisions embodied in them are 'not optimal' and asking us to modify them.

There are several reasons why we have decided not to make such changes to these files.

- However misguided, the current behaviour is clearly what was intended when these classes were designed.
- It is not good practice to change such aspects of 'standard classes' because many people will be relying on them.

We have therefore decided not to even consider making such modifications, nor to spend time justifying that decision. This does not mean that we do not agree that there are many deficiencies in the design of these classes, but we have many tasks with higher priority than continually explaining why the standard classes for LaTeX cannot be changed.

We would, of course, welcome the production of better classes, or of packages that can be used to enhance these classes.

New addresses for TUG

For information about joining the TeX Users Group, and about lots of other LaTeX-related matters, please contact them at their new address:

TeX Users Group, P.O. Box 1239,
Three Rivers, CA 93271-1239, USA
Fax: +1 209 561 4584
E-mail: `tug@mail.tug.org`
URL: `http://www.tug.org/`

LaTeX News

Issue 6, December 1996

Welcome to LaTeX News 6

This issue of *LaTeX News* accompanies the sixth release of the new standard LaTeX, LaTeX 2_ε.

Mono-case file names

Previously LaTeX has used some files with 'mixed-case' file names such as `T1cmr.fd` and `T1enc.def`.

These file names cause problems on some systems (in particular they are illegal on the ISO 9660 CDROM format) and so in this release all file names have been made lowercase (for example `t1cmr.fd` and `t1enc.def`).

This change should *not* affect any document. Within LaTeX, encodings still have the usual uppercase names in uses such as `\usepackage[T1]{fontenc}` and `\fontencoding{T1}`. LaTeX will automatically convert to the lowercase form while constructing the file name. LaTeX will input the 'fd' file under the old name if it fails to find the file with the new name, so existing collections of fd files should still work with this new release.

The change *does* affect the configuration files that may be used to make the LaTeX format with initex. For example, the file `fonttext.ltx` previously specified `\input{T1cmr.fd}`. It now has `\input{t1cmr.fd}`. If you use a local file `fonttext.cfg` you will need to make similar changes, as `\input{T1cmr.fd}` will not work as `T1cmr.fd` is no longer in the distribution.

The files affected by this change all have names of the form `*.fd` or `*enc.def`.

Another input encoding

Thanks to work by Søren Sandmann, the inputenc package now supports the IBM codepage 865 used in Scandinavia.

Better user-defined math display environments

Suppose that you want to define an environment for displaying text that is numbered as an equation. A straightforward way to do this is as follows:

```
\newenvironment{texteqn}
  {\begin{equation}
     \begin{minipage}{0.9\linewidth}}
    {\end{minipage}
  \end{equation}}
```

However, if you have tried this then you will probably have noticed that it does not work perfectly when used in the middle of a paragraph because an inter-word space appears at the beginning of the first line after the environment.

There is now an extra command (with a very long name) available that you can use to avoid this problem; it should be inserted as shown here:

```
\newenvironment{texteqn}
  {\begin{equation}
     \begin{minipage}{0.9\linewidth}}
    {\end{minipage}
  \end{equation}
  \ignorespacesafterend}
```

Docstrip improvements

The docstrip program that is used to unpack the LaTeX sources has undergone further development. The new version should be able to process all old 'batchfiles' but it allows a simpler syntax in new 'batchfiles' (no need to define `\def\batchfile{...}`).

It also allows 'target' directories to be specified when writing files. This directory support is disabled by default unless activated in a local `docstrip.cfg` configuration file. See `docstrip.dtx` for details.

AMS LaTeX update

Since the last LaTeX release in June, the American Mathematical Society have re-issued the 'AMSLaTeX' classes and packages, fixing several reported problems.

Graphics package update

The LaTeX color and graphics packages have been updated slightly, principally to support more dvi drivers, see the readme file in the graphics distribution.

EC Fonts released

The first release of the Extended Computer Modern fonts has just been made. (In January 1997.)

This release of LaTeX does *not* default to these 'ec' fonts as its T1 encoded fonts. By default it will use the 'dc' fonts if the T1 encoding is requested.

As noted in `install.txt` you may run TeX on the install file `ec.ins` *after* unpacking the base distribution but *before* making the LaTeX format. This will produce suitable 'fd' files making LaTeX (including, for the first time, the slides class) use the 'ec' fonts as the default T1 encoded font set.

LaTeX News

Issue 5, June 1996

Welcome to LaTeX News 5

This issue of *LaTeX News* accompanies the fifth release of the new standard LaTeX, LaTeX 2_ε.

Extra possibilities for section headings

Most LaTeX sectioning commands are defined using `\@startsection`. For example, the article class defines:

```
\newcommand\section{\@startsection
 {section}{1}{0pt}{-3.5ex plus-1ex minus-.2ex}%
 {2.3ex plus.2ex}{\normalfont\Large\bfseries}}
```

The last argument specifies the style in which the section heading is to be typeset.

The new feature added at this release is that at the *end* of this argument you may specify a command that *takes an argument*. This command will be applied to the section number and heading. For example, one could use the `\MakeUppercase` command to produce uppercase headings. A package or class file could contain:

```
\renewcommand\section{\@startsection
 {section}{1}{0pt}{-3.5ex plus-1ex minus-.2ex}%
 {2.3ex plus.2ex}{\normalfont\Large\MakeUppercase}}
```

to produce section headings using uppercase medium weight text, rather than the bold text used by article. Note that, like the font choice, the uppercasing applies only to the actual heading (including any automatically generated section number), not to the text as it may appear in the running head or table of contents.

The 'openany' option in the 'book' class

The openany option allows chapter and similar openings to occur on left hand pages. Previously this option only affected `\chapter` and `\backmatter`. It now also affects `\part`, `\frontmatter` and `\mainmatter`.

More font (output) encodings

The font encoding name `T3` has been allocated to the encoding used in the new 256-character IPA fonts (for the phonetic alphabet) produced by Rei Fukui. His package, tipa, gives access to these fonts and should soon be available. (The encoding named `OT3` is the 128-character encoding used in the IPA fonts produced by Washington State University.)

More input encodings supported

The inputenc package now supports the IBM codepage 852 used in Eastern Europe, with the option `[cp852]` contributed by Petr Sojka.

Also, the inputenc package now activates most 'control codes' with ASCII values below 32. Currently none of the encodings in the standard distribution makes use of these positions.

Fixes and improvements

The LaTeX kernel has only had minor changes, apart from `\@startsection` mentioned above. However, some small fixes have been incorporated removing the following problems:

- In tabular and array, previous versions of LaTeX 'lost' the inter-column space from an '`l`'-column, when that column was completely empty.
- Previously, the use of the `\nofiles` command could change the *vertical spacing* in a document. A side effect of fixing this is that when `\nofiles` is used, `\label` puts a blank line in the log file.
- LaTeX often loads fonts 'on demand'. Previously, this could happen inside the argument of an accent command and this would cause the accent to appear in the wrong place.

Changes to the 'tools' packages

- The longtable package now uses a modified algorithm, contributed by David Kastrup, to align the 'chunks' of a table. It is now unnecessary to edit the document to add `\setlongtables` before the final run of LaTeX. In certain cases of overlapping `\multicolumn` entries, the new algorithm will produce better column widths than the old (at the price of extra passes through LaTeX).
- The dcolumn package now has the extra possibility of specifying the number of digits both *before* and after the 'decimal point'. This makes it easy to centre the column of numbers under a wide heading.

New copy of the LaTeX bug database

`http://www.tex.ac.uk/ctan/latex/bugs.html` will soon have links to a copy of the searchable LaTeX bugs database at Mainz (Germany) as well as the original copy at Sussex (England).

LaTeX News

Issue 4, December 1995

Welcome to LaTeX News 4

An issue of *LaTeX News* will accompany every future release of LaTeX. It will tell you about important events, such as major bug fixes, newly available packages, or any other LaTeX news. This issue accompanies the fourth release of LaTeX 2$_\varepsilon$.

LaTeX getting smaller

The last release in June started a trend of LaTeX becoming smaller, we are pleased to announce that this has continued with this release. In particular the experimental 'autoload' version described in `autoload.txt` is much smaller as more parts of LaTeX are autoloaded.

New 'concurrent' docstrip

The time taken to 'unpack' this release from the documented sources should be much reduced (roughly half the time, depending on installation conditions). This is due to an improved version of the docstrip program that has been contributed by Marcin Woliński. This can write up to 16 files at once. The previous version could only write one file at a time which meant that it was very slow when producing many small files from the same source file as the source needed to be re-read for each file written.

New T1 encoded fonts

This year Jörg Knappen has completed a new release of the 'Cork' (T1) encoded Computer Modern fonts: the dc fonts release 1.2.

This release of the dc fonts fixes many bugs (including the missing `?‘` (¿) and `!‘` (¡) ligatures) and improves the fonts in many other ways. It is strongly recommended that you upgrade as soon as possible if currently you are using the old dc fonts, release 1.1 or earlier. The new fonts are available from the CTAN archives, in `tex-archive/fonts/dc`.

The names of the font files are *different*. This does not affect LaTeX documents but *does* affect the installation procedure as it assumes that you have the *new* fonts, and will write suitable 'fd' files for those fonts. If you have not yet upgraded your dc fonts then, after unpacking the distribution, you *must* `latex olddc.ins` to produce 'fd' files for the old dc fonts. This must be done *before* the format is made. Running the test document at `ltxcheck.tex` the end of the installation will inform you if the wrong set of 'fd' files has been installed.

Note that this change does not affect the standard 'OT1' Computer Modern fonts that LaTeX uses by default.

More robust commands

The commands `\cite` and `\sqrt` are now robust.

Although most commands with optional arguments are fragile, as documented, such commands defined using the second optional argument of `\newcommand` and its derivatives are now *robust*.

New Interface to building 'extension' classes

The mechanism provided by `\DeclareOption`, `\ProcessOptions` and `\LoadClass` has proved to be a powerful and expressive means of defining one class in terms of another 'base' class. However there have been some requests to simplify the declaration of the common case where you want the 'base' class to be called with *all* the options that were specified to the extension class. This is now provided by the new command `\LoadClassWithOptions`. A similar command `\RequirePackageWithOptions` is provided for package use. More details of this feature are provided in `clsguide.tex` and `ltclass.dtx`.

More Input Encodings

The experimental `inputenc` package allows a more natural style of input of accented and other characters.

Three new input encodings are now supported.

- `ansinew` the Windows ansi encoding, as used in Microsoft Windows 3.x.
- `cp437de` a variant of `cp437`, which uses ß rather than β in the appropriate slot.
- `next` the encoding used on Next computers.

Further information

For more information on TeX and LaTeX, get in touch with your local TeX Users Group, or the international TeX Users Group, 1850 Union Street, #1637, San Francisco, CA 94123, USA, Fax: +1 415 982 8559, EMail: tug@tug.org. The LaTeX home page is `http://www.tex.ac.uk/ctan/latex/` and contains links to other WWW resources for LaTeX.

LaTeX News

Issue 3, June 1995

Welcome to LaTeX News 3

An issue of *LaTeX News* will accompany every future release of LaTeX. It will tell you about important events, such as major bug fixes, newly available packages, or any other LaTeX news.

June 1995 release of LaTeX

June 1995 sees the third release of LaTeX 2$_\varepsilon$. We are on schedule to deliver a release of LaTeX every six months, in December and June.

In the last *LaTeX News*, we said "we don't expect so much activity in the next six months," which has turned out not to be true!

Additional input encodings

In the last release of LaTeX we distributed a test version of the `inputenc` package which allows the use of input characters other than just a–z and A–Z. The package has proved to be robust, so we are now distributing an expanded version. The new release comes with a number of input encodings:

- `ascii` the standard encoding,
- `latin1` the ISO Western European alphabet,
- `latin2` the ISO Eastern European alphabet,
- `cp437` the IBM codepage 437,
- `cp850` the IBM codepage 850, and
- `applemac` the Apple Macintosh encoding.

These can be used by specifying an option to the `inputenc` package, for example:

```
\usepackage[latin1]{inputenc}
```

The new input encodings are currently being tested, but we don't expect any major changes.

LaTeX getting smaller

In the past releases of LaTeX 2$_\varepsilon$, the amount of memory LaTeX requires has increased, but we are pleased to say that this trend has been reversed. We hope that future releases of LaTeX will continue to get smaller.

For example, on this document, the December 1994 release used 52,622 words of memory, and the June 1995 release uses 51,216 words of memory, which is a 2.7% reduction.

We are currently experimenting with other ways of reducing the size of LaTeX. For example, we are experimenting with an option to remove the `picture` and `tabbing` environments from the LaTeX kernel, and to load them from a file the first time they are used. This should help LaTeX to run on machines with limited memory. See `autoload.txt` for details.

Distribution and modification

One topic of discussion that has kept us busy is the distribution and modification conditions of LaTeX. We are committed to keeping LaTeX as free reliable software, and ensuring that (as far as possible) LaTeX documents will produce the same results on all systems.

The modification conditions are currently under discussion, and we would like to hear from anyone interested. Please read `modguide.tex` for more information.

AMS-LaTeX full release

The AMS-LaTeX packages were still in beta test in the December 1994 release of LaTeX, and the full release came out in January 1995.

AMS-LaTeX is described in the *User's Guide* (`amsldoc.tex`) and in *The LaTeX Companion*.

PostScript fonts

There is a new test release of the PSNFSS packages for accessing PostScript fonts in LaTeX 2$_\varepsilon$. This includes an update to all of the fonts, to remove many of the underfull and overfull `\hbox` warnings, and improve the setting of non-English languages.

The new release of LaTeX removes all of the 'hidden' uses of Computer Modern mathematics. For example, the footnote markers used to use math mode, so always used Computer Modern digits rather than ones from the current text font. This has now been fixed.

Further information

For more information on TeX and LaTeX, get in touch with your local TeX Users Group, or the international TeX Users Group, P. O. Box 869, Santa Barbara, CA 93102-0869, USA, Fax: +1 805 963 8358, EMail: tug@tug.org.

The LaTeX home page is `http://www.tex.ac.uk/ctan/latex/` and contains links to other WWW resources for LaTeX.

LaTeX News

Issue 2, December 1994

Welcome to LaTeX News 2

An issue of *LaTeX News* will accompany every future release of LaTeX. It will tell you about important events, such as major bug fixes, newly available packages, or any other LaTeX news.

December 1994 release of LaTeX

December 1994 sees the second release of LaTeX 2$_\varepsilon$. We are on schedule to deliver a release of LaTeX every six months, in December and June.

This release has seen quite a lot of activity, which is not too surprising as it's only been a year since the first test release of LaTeX 2$_\varepsilon$. We don't expect so much activity in the next six months.

Many of the changes are minor improvements and bug-fixes—see *LaTeX 2$_\varepsilon$ for authors* (`usrguide.tex`), *LaTeX 2$_\varepsilon$ font selection* (`fntguide.tex`) and our change log (`changes.txt`) for more details.

However, there are two important new packages available for LaTeX: `inputenc` and AMS-LaTeX.

Accented input

One of the problems with writing non-English documents in LaTeX is the accent commands. Reading documents containing text like `na\"\i ve` is frustrating, especially if your keyboard allows you to type `naïve`.

In the past, LaTeX has not supported input containing accented characters such as ï, because Windows, Macintosh and Unix all have different ways of dealing with accented input, called *input encodings*.

However, the `inputenc` package allows you to specify which input encoding your document is written with, for example to use the ISO Latin-1 encoding, you type:

```
\usepackage[latin1]{inputenc}
```

At the moment, `inputenc` supports the `ascii` and `latin1` input encodings, but more will be added with future releases.

The `inputenc` package is currently a test release. The user interface for the full release will be upwardly compatible with the test version.

AMS-LaTeX

AMS-LaTeX is a set of miscellaneous extensions for LaTeX distributed by the American Mathematical Society. They provide superior information structure and superior printed output for mathematical documents.

There are far too many features of AMS-LaTeX to list here. AMS-LaTeX is described in the accompanying documentation, and in *The LaTeX Companion*.

Version 1.2beta of AMS-LaTeX was released for testing by intrepid users in October 1994. The full release of AMS-LaTeX 1.2 is expected in early January 1995.

It will be divided into two bundles:

- the `amsfonts` packages, which give access to hundreds of new mathematical symbols, and new math fonts such as blackboard bold and fraktur.
- the `amsmath` packages, which provide finer control over mathematical typesetting, such as multi-line subscripts, enhanced theorem and proof environments, and improved displayed equations,

For compatibility with older documents, an `amstex` package will be provided.

LaTeX on the internet

LaTeX has its own home page on the World Wide Web, with the URL:

```
http://www.tex.ac.uk/CTAN/latex/
```

This page describes LaTeX and the LaTeX3 project, and contains pointers to other LaTeX resources, such as the user guides, the TeX Frequently Asked Questions, and the LaTeX bugs database.

The electronic home of anything TeX-related is the Comprehensive TeX Archive Network (CTAN). This is a network of cooperating ftp sites, with over a gigabyte of TeX material:

```
ftp://ftp.tex.ac.uk/tex-archive/
ftp://ftp.shsu.edu/tex-archive/
ftp://ftp.dante.de/tex-archive/
```

For more information, see the LaTeX home page.

Further information

For more information on TeX and LaTeX, get in touch with your local TeX Users Group, or the international TeX Users Group, P. O. Box 869, Santa Barbara, CA 93102-0869, USA, Fax: +1 805 963 8358, EMail: tug@tug.org.

LaTeX News

Issue 1, June 1994

Welcome to LaTeX News

An issue of *LaTeX News* will accompany every future release of LaTeX. It will tell you about important events, such as major bug fixes, newly available packages, or any other LaTeX news.

LaTeX 2ε—the new LaTeX release

The most important news is the release of LaTeX 2ε, the new version of the LaTeX software. This version has better support for fonts, graphics and colour, and will be actively maintained by the LaTeX3 project team. Upgrades will be issued every six months, in June and December.

Why a new LaTeX?

Over the years many extensions have been developed for LaTeX. This is, of course, a sure sign of its continuing popularity but it has had one unfortunate result: incompatible LaTeX formats came into use at different sites. Thus, to process documents from various places, a site maintainer was forced to keep LaTeX (with and without NFSS), SliTeX, AMS-LaTeX, and so on. In addition, when looking at a source file it was not always clear for which format the document was written.

To put an end to this unsatisfactory situation a new release of LaTeX was produced. It brings all such extensions back under a single format and thus prevents the proliferation of mutually incompatible dialects of LaTeX 2.09. The new release was available for several months as a test version, and the final release of 1 June officially replaces the old version.

Processing documents with LaTeX 2ε

Documents written for LaTeX 2.09 will still be read by LaTeX 2ε. Any such document is run in *LaTeX 2.09 compatibility mode*.

Unfortunately, compatibility mode comes with a price: it can run up to 50% slower than LaTeX 2.09 did. If you want to run your document in the faster *native mode*, you should try replacing the line:

```
\documentstyle[options,packages]{class}
```

with:

```
\documentclass[options]{class}
\usepackage{latexsym,packages}
```

Unfortunately, this will not always work, because some LaTeX 2.09 packages will only work in LaTeX 2ε compatibility mode. You should find out if there is a LaTeX 2ε version of the package available.

LaTeX 2ε native mode also gives access to the new features of LaTeX 2ε, described in *LaTeX 2ε for authors*.

New packages

LaTeX 2ε has much better support for graphics, colour, fonts, and multi-lingual typesetting. The following software should be available from the distributor who brought you LaTeX 2ε:

babel, for typesetting in many languages.

color, for colour support.

graphics, for including images.

mfnfss, for using bitmap fonts.

psnfss, for using Type 1 fonts.

tools, other packages by the LaTeX3 team.

The packages come with full documentation, and are also described in *LaTeX: A Document Processing System* or *The LaTeX Companion*.

Further information

More information about LaTeX 2ε is to be found in:

LaTeX: A Document Preparation System, Leslie Lamport, Addison Wesley, 2nd ed, 1994.

The LaTeX Companion, Goossens, Mittelbach and Samarin, Addison Wesley, 1994.

The LaTeX distribution comes with documentation on the new features of LaTeX:

LaTeX 2ε for authors, describes the new features of LaTeX documents, in the file `usrguide.tex`.

LaTeX 2ε for class and package writers, describes the new features of LaTeX classes and packages, in the file `clsguide.tex`.

LaTeX 2ε font selection, describes the new features of LaTeX fonts for class and package writers, in the file `fntguide.tex`.

For more information on TeX and LaTeX, get in touch with your local TeX Users Group, or the international TeX Users Group, P. O. Box 869, Santa Barbara, CA 93102-0869, USA, Fax: +1 805 963 8358, EMail: tug@tug.org.

Calendar

1997

Jun 3 – 7 Joint International Conference ACH-ALLC'97 (Association for Computers and the Humanities, and Association for Literary and Linguistic Computing), Queen's University, Kingston, Ontario, Canada. For information, visit `http://www.qucis.queensu.ca/achallc97`.

Jun 4 – 6 SSP 19th Annual Meeting, Society for Scholarly Publishing, JW Marriott Hotel, Washington, DC. For information, visit `http://www.edoc.com/ssp/` or write to `ssp@resourcenter.com`.

Jun 5 DANTE TEX–Stammtisch at the Universität Bremen, Germany. For information, contact Martin Schröder (`MS@Dream.HB.North.de`; telephone +49-421-2239425). *Regular schedule*: First Thursday (if not a holiday), 18:00, Universität Bremen MZH, 28359 Bremen, 4th floor, across from the elevator.

Jun 5 DANTE TEX–Stammtisch at the Universität Karlsruhe, Germany. For information, contact Klaus Braune (`braune@rz.uni-karlsruhe.de`; telephone 0721/608-4031). *Regular schedule*: First Thursday (if not a holiday), 19:30, Rechenzentrum der Universität Karlsruhe, Zirkel 2, 3.0G Raum 316.

Jun 12 NTG 19th Meeting, Technische Universiteit Delft, The Netherlands. For information, contact `ntg@nic.surfnet.nl` or visit `http://www.ntg.nl/NEW/bijeen/bijeen19.html`.

Jun 26 DANTE TEX–Stammtisch, Berlin, Germany. For information, contact Rolf Niepraschk (`niepraschk@ptb.de`). Last Thursday, 19:00, Gaststätte "Bärenschänke" Friedrichstr. 124 near the U-Bahnhof "Oranienburger Tor".

Jul 3 DANTE TEX–Stammtisch at the Universität Bremen, Germany. (For details, see Jun 5.)

Jul 4 – 7 SHARP Conference (Society for the History of Authorship, Reading and Publishing), University of Cambridge, U.K. For information, contact James Raven, SHARP Conference Programme Committee, 51 Sherlock Close, Cambridge CB3 0HP, U.K.

Jul 28 – Aug 1 **TUG'97** — The 18th annual meeting of the TEX Users Group, "*TEX Comes Home*", held at the Lone Mountain Conference Center in San Francisco, CA, USA.

Sep 22 – 23 LAMPE 97: Lausanne — Atelier sur les Modèles de Page Électronique/ Workshop on Electric Page Models, EPFL, Lausanne, Switzerland. For more information, visit `http://www.irisa.fr/lampe97`.

Sep 25 – 26 Fourth international conference, Hypertexts and Hypermedia: Products, Tools, Methods, Université de Paris VIII, Laboratoire Paragraphe, Paris, France. For information, contact Imad Saleh or Alain Lelu, (`conf97@labart.univ-paris8.fr` or `lelu@cnam.fr`). Submissions due by 2 Apr 1997.

1998

Apr 1 – 3 EP 98: Seventh International Conference on Electronic Publishing, Document Manipulation and Typography, St. Malo, France. For information, visit `http://www.irisa.fr/ep98`. or contact Jacques André (`jandre@irisa.fr`). Submissions (full papers) due by 15 Jul 1997.

Jun 3 – 5 SSP 20th Annual Meeting, Society for Scholarly Publishing, San Diego, California. For information, visit `http://www.edoc.com/ssp/` or write to `ssp@resourcenter.com`.

For additional information on TUG-sponsored events listed above, contact the TUG office (209-561-0112, fax: 209-561-4584, e-mail: `tug@mail.tug.org`). For events sponsored by other organizations, please use the contact address provided.

Status as of 31 May 1997

TEX Comes Home

18th Annual Meeting of the TEX Users Group

Preliminary Program

Date	*Description / Speaker*
Saturday	**1:00–4:00 Board of Directors Meeting**
July 26, 1997	
Sunday	**Preconference Workshops**
July 27, 1997	**Registration 1:00–5:00**
10:00–3:00	**Board of Directors Meeting**
2:00–5:00	Aspects of Omega; from everyday use to development and extension of multilingual tools / *Yannis Haralambous & John Plaice*
2:00–5:00	LaTeX2HTML / *Ross Moore*
2:00–5:00	Moving from LaTeX 2.09 to LaTeX 2ε / *Anita Hoover*
	Welcome Reception 6:00–8:00 *Join us for an informal get-together*
Monday	**Pictures and TEX**
July 28, 1997	**7:00 a.m.–4:00 p.m. Daily Registration**
9:00–9:45	Opening Convocation /*Michel Goossens, President*
9:45–10:00	Break
	Picture this: the TEXxie approach to graphical illustration
10:00	"Xy-pic as a tool for VHL2G and how this made TEX into an animation tool" / *Kristoffer H. Rose*
10:30	"A tutorial on MetaPost graphs" / *Sebastian Rahtz*
11:00	"Drawing with DraTEX" / *Eitan Gurari*
11:30	"High-quality lables on included graphics, using Xy-pic" / Ross Moore
12:00–1:30	Lunch
	Tooling up: where are we with TEX?
1:30	"The state of ε-TEX" /*ε-TEX member*
2:00	"Omega, the full release" / *John Plaice & Yannis Haralambous*
2:30	"TEX Live 2 – towards a fully flexible TEX on CD-ROM" / *The TEX Live Team*
3:00	Break
3:30	"New font tools for TEX" / *Werner Lemberg*
4:00	"Production of complicated and highly interactive documents" / *Hans Hagen*
Tuesday	**The Web and SGML**
July 29, 1997	**7:00 a.m.–4:00 p.m. Daily Registration**
8:30–9:45	**TUG Business Meeting**
	TEX and scientific publishing on the Internet
10:00	"A new TEX math font family for Elsevier" / *Yannis Haralambous*
10:30	"DVIPDF and graphics" / *Sergey Lesenko*
11:00	"TEX to PDF direct" / *Han The Thanh*
11:30	"Developments in PDF, and LaTeX" / *T.V. Raman*
12:00–1:30	Lunch
1:30	"techexplorer: Interactive scientific electronic publishing for the Internet" / *R.S. Sutor, A.L. Diaz and S.S. Dooley*
2:00	"Translating SGML to HTML, with help from TEX" / *Chris Hamlin*

– – – Continued – – –

*Visit WKPIX, 'San Francisco LIVE' – `http:www.wpix.com` – for more photographs like the one above!

Preliminary Program *continued*

Date	*Description / Speaker*
	TEX behind the scenes: what is our relationship to SGML?
2:30	"The DSSSL style sheet language" / *Jon Bosak*
3:00	"The TEX backend for Jade" /*Sebastian Rahtz*
3:30–4:00	Break
4:00	"DSSSL, TEX and math" / *Chris Rowley*
4:30	"LATEX2HTML – past, present and future" / *Ross Moore*
Wednesday	**Publishing and TEX**
July 30, 1997	**7:00–4:00 Daily Registration**
	TEX and the real world
8:30–9:30	Panel Discussion
9:30–10:00	Break
10:00	"Lessons they should have learned from *The LATEX Graphics Companion*" / *Art Ogawa*
10:30	"Working with process color and TEX" / *Dan Olson*
11:00	"Custom legal documents for the Auto Loan Exchange" / *Douglas Lovell*
11:30	"What LATEX needs to make it useful to publishers" / *Fred Bartlett*
12:00–1:30	Lunch
1:30	Vendor presentations
	LATEX — state of the art?
2:30	"Developments in LATEX for multilingual documents" / *Frank Mittelbach*
3:00	Break
3:30	"Rewriting LATEX's math" / *Michael Downes*
4:00	Discussion of LATEX
Thursday	**LATEX, Fonts and Languages**
July 31, 1997	
	Real Work
8:30	"LATEX as markup for electronic delivery of courseware" / *Mimi Jett*
9:00	"TEX meets watermark" / *Kazuhiro Kitagawa*
	Birds of a Feather
10:00–12:00	BOFs (*concurrent*)
12:00–1:30	Lunch
	Multilingual typography without boundaries
2:00	"The CJK package: multilingual support beyond Babel" / *Werner Lemberg*
3:00	"The multilingual interface of the ConTEXt macro package" / *Hans Hagen*
3:30	**Closing ceremonies**
Friday	**9:00-1:00 Board of Directors Meeting**
August 1, 1997	

Registration cost includes coffee/tea breaks, lunch for five days, and the Wednesday evening banquet. (Breakfasts are included with lodging listed below. Evening meals other than the banquet are not covered by the registration fee.) Computer accounts will be given out to all conference attendees. (Amounts are in U.S. Dollars.)

TUG/LUG members – $360.00; Nonmembers – $420.00; Single day admission – $150.00

Accommodations Economical guest housing on campus is available to conference attendees. Shuttle transportation for those attendees staying on campus is provided to the conference center. Lodging is for Saturday night (July 26)–Thursday night (July 31) and includes breakfast. (Amounts are in U.S. dollars.)

Single occupancy – $240.00; double occupancy – $210

Facilities The Lone Mountain Conference Center is located on the University of San Francisco campus at 2130 Fulton Street, near the northeast corner of Golden Gate Park. From its 55-acre hilltop refuge, it boasts views of the Pacific Ocean, San Francisco Bay, and the dramatic downtown skyline. The center features several lecture halls and dining facilities.

Visit our WWW site (`http://tug.cs.umb.edu/tug97`) for program updates and to register electronically! Or, contact the TUG office at the address inside the front cover.

Institutional Members

American Mathematical Society,
Providence, *Rhode Island*

CNRS - IDRIS,
Orsay, *France*

CERN, *Geneva*, *Switzerland*

College of William & Mary,
Department of Computer Science,
Williamsburg, *Virginia*

Communications
Security Establishment,
Department of National Defence,
Ottawa, *Ontario*, *Canada*

CSTUG, *Praha*, *Czech Republic*

Elsevier Science Publishers B.V.,
Amsterdam, *The Netherlands*

Florida State University,
Supercomputer Computations
Research, *Tallahassee*, *Florida*

Hong Kong University of
Science and Technology,
Department of Computer Science,
Hong Kong

Institute for Advanced Study,
Princeton, *New Jersey*

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

Iowa State University,
Computation Center,
Ames, *Iowa*

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

Marquette University,
Department of Mathematics,
Statistics and Computer Science,
Milwaukee, *Wisconsin*

Masaryk University,
Faculty of Informatics,
Brno, *Czechoslovakia*

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

New York University,
Academic Computing Facility,
New York, *New York*

Nippon Telegraph &
Telephone Corporation,
Basic Research Laboratories,
Tokyo, *Japan*

Princeton University,
Department of Mathematics,
Princeton, *New Jersey*

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

Space Telescope Science Institute,
Baltimore, *Maryland*

Springer-Verlag,
Heidelberg, *Germany*

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

University of California, Berkeley,
Center for EUV Astrophysics,
Berkeley, *California*

University of California, Irvine,
Information & Computer Science,
Irvine, *California*

University of Canterbury,
Christchurch, *New Zealand*

University College,
Cork, *Ireland*

University of Delaware,
Computing and Network Services,
Newark, *Delaware*

Universität Koblenz–Landau,
Koblenz, *Germany*

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

University of Stockholm,
Department of Mathematics,
Stockholm, *Sweden*

University of Texas at Austin,
Austin, *Texas*

Università degli Studi di Trieste,
Trieste, *Italy*

Uppsala University,
Uppsala, *Sweden*

Vrije Universiteit,
Amsterdam, *The Netherlands*

Wolters Kluwer,
Dordrecht, *The Netherlands*

Yale University,
Department of Computer Science,
New Haven, *Connecticut*

TEX Consulting & Production Services

Information about these services can be obtained from:

TEX Users Group
P. O. Box 1239
Three Rivers, CA 93271-1239, U.S.A.
Phone: +1 209 561-0112
Fax: +1 209 561-4584
Email: tug@mail.tug.org

North America

Anagnostopoulos, Paul C.
Windfall Software,
433 Rutland Street, Carlisle, MA 01741;
(508) 371-2316; greek@windfall.com
We have been typesetting and composing high-quality books and technical Publications since 1989. Most of the books are produced with our own public-domain macro package, ZzTEX, but we consult on all aspects of TEX and book production. We can convert almost any electronic manuscript to TEX. We also develop book and electronic publishing software for DOS and Windows. I am a computer analyst with a Computer Science degree.

Cowan, Dr. Ray F.
141 Del Medio Ave. #134, Mountain View, CA 94040;
(415) 949-4911; rfc@netcom.com
Twelve Years of TEX and Related Software Consulting: Books, Documentation, Journals, and Newsletters
TEX & LATEX macropackages, graphics; PostScript language applications; device drivers; fonts; systems.

Hargreaves, Kathryn
135 Center Hill Road, Plymouth, MA 02360-1309;
(508) 224-2367; letters@cs.umb.edu
I write in TEX, LATEX, METAFONT, MetaPost, PostScript, HTML, C, and sometimes C++. I take special care with mathematics. I also copyedit, proofread, write documentation, do spiral binding, scan B/W images at 1200 DPI, program, hack fonts, and design letterforms, ads, newsletters, journals/proceedings and books. I'm a journeyman typographer and began typesetting and designing in 1979. I coauthored *TEX for the Impatient* (Addison-Wesley, 1990) and psychophysics research papers. I have an MFA in Painting/Sculpture/Graphic Arts and an MSc in Computer Science. I'm currently doing some digital type and human vision research.

Hoenig, Alan
17 Bay Avenue, Huntington, NY 11743; (516) 385-0736
TEX typesetting services including complete book production; macro writing; individual and group TEX instruction.

NAR Associates
817 Holly Drive E. Rt. 10, Annapolis, MD 21401;
(410) 757-5724
Extensive long term experience in TEX book publishing with major publishers, working with authors or publishers to turn electronic copy into attractive books. We offer complete free lance production services, including design, copy editing, art sizing and layout, typesetting and repro production. We specialize in engineering, science, computers, computer graphics, aviation and medicine.

Ogawa, Arthur
40453 Cherokee Oaks Drive,
Three Rivers, CA 93271-9743;
(209) 561-4585
Experienced in book production, macro packages, programming, and consultation. Complete book production from computer-readable copy to camera-ready copy.

Quixote Digital Typography, Don Hosek
555 Guilford, Claremont, CA 91711;
(909) 621-1291; Fax: (909) 625-1342;
dhosek@quixote.com
Complete line of TEX, LATEX, and METAFONT services including custom LATEX style files, complete book production from manuscript to camera-ready copy; custom font and logo design; installation of customized TEX environments; phone consulting service; database applications and more. Call for a free estimate.

Richert, Norman
1614 Loch Lake Drive, El Lago, TX 77586;
(713) 326-2583
TEX macro consulting.

Southern California PrintCorp
(800) 899-7267 x801
SAVE MONEY on 1-day Linotronic output for journals. Special *TUGboat* offer. Call now for more information.

Southern California PrintCorp
1915 Midwick Drive, Suite B, Altadena, CA 91001
(800) 899-7267 x888, Fax (818) 399-3565,
BBS (818) 398-3567
We have a ten year history providing 24-hour turn-around imagesetting of PostScript files. Call for FREE information on how *TUGboat*-ers can obtain low-cost, fastest available Linotronic publication production services in the U.S.

Type 2000
16 Madrona Avenue, Mill Valley, CA 94941;
(415) 388-8873; Fax: (415) 388-8865
pti@crl.com
$2.50 per page for 2000 DPI TEX and PostScript camera ready output! We provide high quality and fast turnaround to dozens of publishers, journals, authors and consultants who use TEX. Computer Modern, PostScript and METAFONT fonts available. We accept DVI and PostScript files only and output on RC paper. $2.25 per page for 100+ pages, $2.00 per page for 500+ pages; add $.50 per page for PostScript.

Outside North America

TypoTEX Ltd.
Electronical Publishing, Battyány u. 14. Budapest, Hungary H-1015; (036) 11152 337
Editing and typesetting technical journals and books with TEX from manuscript to camera ready copy. Macro writing, font designing, TEX consulting and teaching.