TUGboat

Volume 20, Number 1 / March 1999

General Delivery

Language Support

TeXLive CD-ROM

Software & Tools

LaTeX

Hints & Tricks

Abstracts

News & Announcements

Late-Breaking News

TUG Business

Advertisements

In old days, men had the rack. Now they have the Press. That is an improvement, certainly. But still it is very bad, and wrong, and demoralising.

Oscar Wilde [1891]
The Soul of Man (1912)

TUGBOAT

COMMUNICATIONS OF THE TEX USERS GROUP
EDITOR BARBARA BEETON

VOLUME 20, NUMBER 1 • MARCH 1999
PORTLAND • OREGON • U.S.A.

TUGboat

During 1999, the communications of the TeX Users Group will be published in four issues. The September issue (Vol. 20, No. 3) will contain the Proceedings of the 1999 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. 20, No. 2. The deadline for technical items will be May 24; reports and similar items are due by June 1. Mailing is scheduled for June. Deadlines for other future issues are listed in the Calendar, page 65.

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, or to the Production Manager, Mimi Burbank (see addresses on p. 3).

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. Send e-mail to `TUGboat@tug.org`, or write or call the TUG office.

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@tug.org` or to the Editor, Barbara Beeton (see address on p. 3).

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

Production Team:

See page 3 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 at `tug-pub@tug.org` or 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.
techexplorer is a trademark of IBM Research.
TeX and 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.

TeX Online:
Untangling the Web and TeX

August 15–19, 1999

University of British Columbia
Vancouver, BC Canada

“Oh, what a tangled web is TeX”

— Ross Moore
(with apologies to Sir Walter Scott [1771-1832],
“Oh, what a tangled web we weave. . . ”
Marmion, canto vi, stanza 17)

Discover your **W**it, **W**isdom, and **W**ay with words in verse on the TUG’99 theme.

TUG’99 Poetry Contest

Contest Guidelines

The aim is to develop the following verse into a full poem, complete and worthy for publication in that illustrious journal, *TUGboat*.

> ***Oh, what a tangled web is TeX,***
> *or so it seems at the outset;*
> *for highest quality, the best to look,*
> *Oh why did I choose to typeset my own book!*
> . . .
> . . .

Submit new verses, single lines or even a whole separate poem, which embody the trials, tribulations, joys and euphoria of learning/using/mastering the typesetting power of TeX or LaTeX.

Stand-alone poems should include a **tangled web** quotation (either variation, implied or exact) or the **www** online theme, somewhere in the work.

Book worm poets may use other tangled web quotations in the literature. Or see the bard’s latest scribblings; help him get some more ideas . . .

`http://www.tug.org/tug99/poetry/contest`

All submissions will be considered, with the best and cleverest verses, lines and phrases being incorporated into the final masterpiece. This will be read at an appropriate meal-time during the TUG’99 conference, and published in the proceedings issue of *TUGboat*. Authors present will receive an appropriate reward for their contribution.

Meta-poetry will also be accepted in the form of graphics, style-files, special fonts, packages for annotations or acknowledgements (or apologies). Submit whatever could be appropriate or useful.

Entries should be submitted by **Thursday, July 15, 1999**, to

`tug99-poetry@tug.org`

but sufficiently witty late entries may be accepted, at the discretion — or state of inebriation — of the judging panel.

General Delivery

From the Vice President

Kristoffer H. Rose

Greetings, fellow TUG members!

We start 1999 as we did 1998 with a *TUGboat* loaded with a fresh edition of TUG's "**TEX Live**" CD for our members. This is the fourth version in just as many years; thus we are converging towards making yearly distribution a tradition. To emphasize this we have followed another TEX tradition and endorsed the CD with original artwork by Duane Bibby. This way we hope that you will proudly show it to your friends and explain to those few of them that are not already TUG members why they should join up to also get this unique piece that may become a collectors item one day – who knows?

While you are waiting for the CD to become a prestigious coaster it has lots of more direct advantages: it has runnable TEX systems for all major Unix variants (including Linux), Windows systems (NT, 95, and 98), for Macintosh, DOS, and OS/2. And of course it contains the latest versions of LATEX and a very large number of useful macro packages as well as full support for producing your documents in the world-wide web formats HTML and Adobe PDF. We are extremely grateful to Sebastian Rahtz, the editor of the **TEX Live** series, for the huge amount of work he has put into the integration of such a large collection of useful stuff!

Just one admonitory remark: each package on the CD comes with its own license. Please respect them so we can continue to provide you with a complete product. (We agree that this is a hassle and we are working on identifying a usable subset of **TEX Live** without usage or redistribution constraints — the CTAN archive maintainers have started a similar effort for their collections.)

But first of all: pop the CD in your drive, read the documentation in the `tldoc` directory, install what you want, and typeset beautiful documents!

Otherwise you will find in this issue an elaborate invitation to TUG '99 to be held in Vancouver in British Columbia (Canada) this summer. You can check for yourself that the organizers have a great program that you will want to be entangled in. A month later the EuroTEX conference in Heidelberg takes up "paperless TEX" — the program for that is just being assembled as these lines are being written.

Also you will find a description of the members of the newly appointed TUG board of directors in this issue. We welcome two new members of the board, Cheryl and Stephanie, whom you can read about in those pages. As always the board is committed to further the purpose of TUG:

> "To provide an organization for those who have an interest in systems for typesetting technical text and font design; to exchange information of same and associated use of computers and computer peripheral equipment; to establish standards and provide channels to facilitate the exchange of macro packages, etc., through publications and otherwise; and to develop, implement and sponsor educational programs, seminars, and conferences in connections with the foregoing. . . "

In the future this will involve the growing number of standards derived from SGML. In general SGML processing TEX has already proven it's value: the only completely free SGML processing system that supports the DSSSL formatting standard is the combination of James Clark's "Jade" processor using LATEX with Sebastian Rahtz's jadetex package as a back-end. TEX made it possible to create a system that produces PDF directly from SGML! Specifically, standards for "typesetting technical text" are only just emerging, however, with the recent "MathML" proposal seeming to take off. If we want TEX to play a role in the future we must invest time and effort in learning about the requirements of these systems and to what extent TEX is suitable or adaptable for the job. I hope the articles and information in this issue will inspire you to start thinking about how we can best approach it.

◇ Kristoffer H. Rose
LIP, ENS-Lyon
46, alle d'Italie
69364 Lyon 7 (France)
`krisrose@tug.org`

Editorial Comments

Barbara Beeton

New book by Don Knuth

One fine March day, I arrived at my desk to find a package from Stanford's Center for the Study of Language and Information (CSLI). Unwrapping it eagerly, I found a copy of *Digital Typography*, inscribed to me by Don. This collection brings together most of Don's short writings on typography. The only TEX-related paper that I miss is "The Errors of TEX", but that has already appeared in a collection, *Literate Programming*, also published (in 1992) by CSLI; however, there's a chapter "The Final Errors of TEX", which analyzes the types of bugs found and fixed since 1988, and also summarizes some unfortunate "features" and several design flaws that he would have changed had he "thought of them early enough before deciding to finalize the design."

Much of the content of *Digital Typography* has appeared in *TUGboat* over the years; of 34 articles, 19 appeared first in *TUGboat*, and one more, which first appeared in the NTG MAPS, was also published here. Going through the pages has given me a superb overview of the history of TEX and also of my experiences as editor of *TUGboat*.

A review or two should be available for the next issue. The book is introduced on Don's Web pages, along with a list of its glitches; don't bother to report the absence of running heads on pages 361, 363 and 365 — I've already claimed that, and received my to-be-framed-rather-than-cashed check for $2.56 along with the explanation of how it happened. (For the curious, the article, "AMS Euler — A New Typeface for Mathematics", was full of special active characters, enclosed in

```
\begingroup \catcode ... \endgroup
```

and the chapter began inside of the group. "The CSLI format set `\chaptertitle` for running heads, and `\chaptertitle` became vacuous when the group ended.")

While you're waiting for the next issue and the reviews, check it out at the bookstore or the library if you get a chance. The ISBN is 1-57586-011-2.

ε-TEX news

Peter Breitenlohner announced the release of a new version (2.1) of ε-TEX. This release fixes a bug in the handling of marks for implementations where `null=min_halfword` is not zero.

ε-TEX work will no longer be carried out within the $\mathcal{NTS}$ working group, but will proceed on its own under Peter Breitenlohner's direction; the rest of the ε-TEX team will continue this work as well.

We will try to obtain reports from both the ε-TEX and $\mathcal{NTS}$ groups for the next issue.

Patent for style sheets in electronic publishing for Microsoft

On 12 January 1999, patent number US5860073 was issued to Microsoft Corporation for "Style sheets for publishing system". The abstract reads as follows:

> The use of style sheets in an electronic publishing system is described. A style sheet is a collection of formatting information, such as font and tabs in a textual document. The style sheets described herein are applied to individual display regions (controls) on a page. Unlike previous systems, the display regions in this system do not contain any text at the time the style sheet is applied. Rather, the text, or other media such as graphics, is poured into the display region when the title is rendered on the customer's computer.[1]

While the references attached to this citation pertain specifically to "cascading style sheets", the title and abstract indicate no such limited scope. Let's hope that the unadorned term "style sheets" doesn't get trademarked like "Kleenex", so that we can continue to use it in other appropriate contexts.

New LATEX Project Public License

In preparation for **TEX Live 4**, which is shipped with this issue, an extraordinary effort was made to regularize the licensing terms under which various packages on the CD are distributed. One of the results of this effort is the restatement of the license for the base LATEX distribution, in a separate document, the LATEX Project Public License (LPPL), which can be cited by other software, even if it is unrelated to TEX.

The LPPL can be found at `ftp://ftp.ctan.org/tex-archive/macros/latex/base/lppl.txt` and equivalent areas at other CTAN sites.

In brief, the license requires that a licensed item carry a copyright notice that cites the LPPL and be accompanied by a list of files covered by the license. It provides for free use of the item as long as

- all files are included in any (re)distribution;
- any file which is to be changed must be renamed before changes are made;
- any modified items are clearly identified, and the address to which error reports are to be sent is changed;

[1] `http://www.patents.ibm.com/patlist?icnt=US&patent_number=5860073`

- the unmodified version of the file, or sufficient equivalent information, is distributed so that the user of the modified version can obtain an original, unmodified version.

All restrictions are intended to target the goals of reliability, portability, and predictability.

Once the LPPL text had been packaged in a separate, easily identified file, the **TEX Live** team encouraged all contributers of packages to the CD to subscribe to it. While the CD does contain some items with other licenses (e.g. some shareware implementations of TEX, included for convenience), nearly all fonts and macro packages and collections now cite the LPPL.

Packages which formerly were released with various restrictions but have now been issued under the LPPL include the Bakoma fonts.

New goodies on CTAN

A number of interesting things have been uploaded to CTAN over the past few months. Here are a few that I found especially intriguing.

- The `CMCYR` fonts in Type 1 format. These are Russian extensions for the Computer Modern fonts. Both METAFONT sources and the Type 1 files are included. From Basil Malyshev. Found in `fonts/cyrillic/cmcyr`.
- Support for typesetting of Cree/Inuktitut in Canadian Aboriginal Syllabics. Includes METAFONT source, a LATEX style package, and a user's manual in LATEX source and PostScript. From Ivan Derzhanski. Found in `language/casyl`.
- The ConTEXt system, by Hans Hagen. (See the articles by Hans in *TUGboat* **19**(3) for a few features of ConTEXt.) Found in `macros/context/` and subdirectories. The documentation is still available only via the Web at `http://www.ntg.nl/context/`.
- The `TeXemplar.cls` class for the new journal of CervanTEX, the Spanish TEX User's Group, which has just been formally registered. From Javier Bezos. Found in `macros/latex/contrib/other/TeXemplar`.

In this issue, a few small LATEX utility packages, some of which have also recently appeared on CTAN, are explored in *The Treasure Chest*.

Welcome to CervanTEX

As mentioned above, there is a newly established language-based group — CervanTEX, also known as GUTH, the Grupo de Usuarios de TEX Hispanohablantes, for Spanish-speaking TEX users.

According to their home page, `http://gordo.us.es/Actividades/CervanTeX/CervanTeX.html`, they are planning their first meeting, in Madrid, 13–14 September 1999. Many other interesting items are mentioned, in particular a large body of documentation in Spanish.

We wish them well.

Electronic *TUGboat*

For a long time, our intention has been to make *TUGboat* available in electronic form, perhaps with some months' delay. That this hasn't happened yet, that the delay sometimes seems to be permanent, is largely a consequence of lack of time and resources. Remember, please, that everyone responsible for putting together *TUGboat* is a volunteer — we all have "other" jobs, some of us have families, and we do like to eat and sleep occasionally.

Another small difficulty is that we have never received permission from authors to distribute their words in any form but print; we assume that submission of an item carries with it permission for it to appear in *TUGboat*, but for anything else, we must ask — and once again, we haven't had time to do that, but we're starting.

For the foreseeable future, we will have to limit what is posted to PDF files; these can be generated nearly automatically from the PostScript files used to generate an issue, so this is the most efficient method available to the production staff. We realize that this may not be convenient for some people; however, the Acrobat reader *is* available for most current platforms, and supporting other formats is simply too time-consuming to make it practicable at this time.

We are making an effort to ask for the necessary permissions, and PDF files are now created for each article as it is completed. Some articles *have* already been posted on the TUG Web pages — they are linked from the tables of contents, which *are* complete.

We invite you to look at the *TUGboat* Web pages and send us your comments. Start at `http://www.tug.org` and follow the *TUGboat* links. If there are particular articles you would like to see appear with some priority, let us know; we will try to work first on items that we know are in demand.

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

Father Truchet, the typographic point, the *Romain du roi*, and tilings

Jacques André and Denis Girou

Abstract

Father Sébastien Truchet (1657-1729) is genuinely recognized as a mathematician (especially for "Truchet tilings"); however, very few typographers know that he is the real inventor of the typographic point or even that he designed the famous *Romain du roi*, which could be considered the first digital font!

Introduction

This paper is adapted from a page on the Web[1] that was created last October for the ATypI conference held in Lyons, France. For us, this was a good opportunity to restore Father Sébastien Truchet's good name. He was born in Lyons in 1657. Even if he is genuinely recognized as a mathematician (especially for "Truchet tilings"), as an expert in hydraulics (he designed most of the French canals) and as an inventor (he invented a fantastic number of things such as sundials, guns, engines to transplant adult trees — they have been used in the Parc de Versailles, etc.) [8, 12], very few typographers know that he is the genius behind the typographic point or even that he designed the famous *Romain du roi* font.

Here are three stories about him.

Figure 1: Father Sébastien Truchet, 1657–1729; after Lery [12].

Sébastien Truchet and the typographic point

In typography at the end of the 17th century, body sizes were not measured but were given names instead (just as today's bold typefaces are referred to as "heavy", "light", or "demi bold" without actually being measured). These names were often based on the title of books in which the characters were used for the first time. For example, "Cicero" was the name given to the body sizes of characters used to print Cicero's *Epistles*, while the characters known as "Saint Augustin" were used in the book *City of God*, written by this Father of the Church. Names could also derive from their appearance: the English "pica" comes from the Latin *pica* ('magpie') since the first printed book used sharply contrasted blacks and whites, like the plumage of the magpie bird. Other names stem from the origin of the characters (e.g., "Parisienne"), their function (French "Canon", English "Brevier" or German "Missal"), or from their relative size ("Petit Romain", "Gros-Canon"), since there was an implicit scale between these sizes (a "little Canon", for example, was twice as large as a "Saint Augustin"). However, sizes differed widely from one foundry to another and it was very difficult to mix types produced by different foundries — or even by a single foundry, for that matter.

The end of the 17th century also coincided with the Age of Enlightenment. Envious of the achievements of foreign countries, Colbert, a minister to Louis XIV, decided in 1692 to prepare a compendium on existing arts, crafts and trades, and appointed four scientists to head this work: Abbot Bignon, Jacques Jaugeon, Gilles Filleau des Billettes and Father Sébastien Truchet. They began work and quickly completed several illustrated descriptions of existing trades and professions. In 1699, Louis XIV elected them to the Academy where they continued their work under the direction of Réaumur. However, in 1750 (Truchet having died by this time), even though no volume of the *Description des Métiers* had yet been published, Diderot

[1] In both French and English: `http://www.irisa.fr/faqtypo/truchet/truchet.html`

and D'Alembert announced the release of their *Encyclopédie*. This vexed the Academy, which then published seventy-three volumes and some two thousand printed plates. Complete sets of the *Description des Métiers* are very rare and this work is quite unknown.

Let us return to the year 1693, when the Bignon Commission undertook its compendium of trades and professions, beginning with: "that art which will preserve all others" — printing.[2]

Members of the commission, in particular Truchet, studied the known characters of the day (even using the recently invented microscope) and proposed three successive systems to quantify types in which a series of body sizes were defined in regular increments, based on the "line", a unit of measure used by silversmiths and equivalent to $^1/_{12}$ of the official inch.[3] Units of length then in use included the *toise*, which comprised two *aulnes* (roughly equivalent to today's metre) which comprised 3 feet; the foot (0.32484 m) was divided into 12 inches (2.707 cm); each inch was divided into 12 lines (of 2.256 mm) and each line into 12 metric points (of 0.188 mm). The problem was that these values varied from place to place. The values given here are known as king ("roi") values. To complicate matters further, the English feet and inches were slightly different!

The initial plan was conceived in 1694, based on several principles:

- Typefaces are measurable. Their measurement is based on a legal unit of length.

- The range of possible body sizes is governed by the following rule: sizes are the result of adding an increment which follows a geometric progression; these are 7.5 9 10.5 12, then 15 18 21 24, and so on. The real size of the type is 7.5 *lignes secondes*,[4] and so on.

The second rule is given below in modern mathematical terms:

- Type sizes are expressed in multiples of c_i of the second line of the "king's inch" (i.e., $^1/_{12} \times {}^1/_{12}$ of an inch, today's 0.1879583 mm).

- The body sizes are defined as follows:
  $C_0 = 7.5$
  $C_i = C_{i-1} + 1.5 \times 2^{i \div 4}$

The second plan, in 1695, was based on $^1/_{24}$ of a line, while the third scheme used a far smaller unit, $^1/_{204}$ of a line, the equivalent of today's 0.011057 mm — or roughly 2,300 dpi! While this value is one to make us dream, it continues to amuse punchcasters since no mechanical tool is able to provide such high precision. Fournier [6, 14], 50 years later, used 7 parts whereas Truchet proposed 204, declaring that "these rules refer to the concept of infinitely small characters that only the imagination can attain! ... Consequently, are so many squares necessary to form an O shape which is round?". The surprising choice of 204 is perhaps related to the product of 12 × 17 where 12 is the duodecimal base and where 17 might be equal to the number of lines (16) encountered in the lowercase designs cast by Simonneau plus $^2/_2$ (two half lines for the shoulders).

However, this proposal had little immediate result — perhaps it was too "academic". All existing punches would have had to be re-cast and all types re-minted! In addition, since the system that Truchet proposed was not totally identical to the one in use at that time (that is, his system did not accurately reflect the less-than-perfect reality of the day), Truchet devised new names for character sizes, such as "la Petite Royale" (body size 12) or "Le Bourbon" (body size 36). Moreover, members of the Chambre syndicale des imprimeurs (Printers' Guild) probably had no knowledge of this research since at their request the king issued a decree on 28 February 1723 which laid down the type heights as well as the official relationships between the old names.

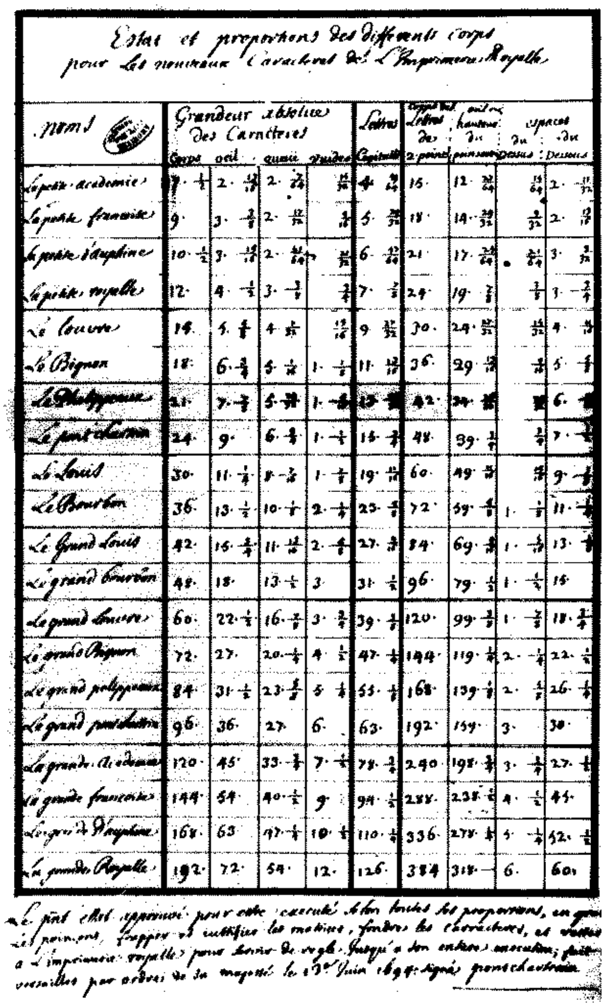
Pierre-Simon Fournier (nicknamed "the Younger") was familiar with the work of Truchet (since he had ridiculed it) and applied the concept of proportions but adapted the scale of typefaces to existing character body sizes. This gave rise to the Fournier point of 1737 which, unfortunately, is not based on a legal unit! Note that the Fournier point was closer to the pica than to the Truchet point.

Didot in turn incorporated Truchet's idea and standardised the typographic point in 1783 (based on $^1/_{72}$ of the French inch but with practically the same progressions as Truchet). As would have been the case with Truchet's typographic point, the punches had to be re-cast, which explains why many printers continued to use Fournier's point until the 20th century. See Boag [2] for more on the history of the point.

[2] From *Histoire de l'Académie royale des sciences... avec les mémoires*, p. 117 et seq., quoted in Jammes [10, p. 6].

[3] We're only in the 17th century and the metric system is still unknown, even though, in 1670, Father Mouton had already proposed to the Academy the *virga*, a unit of length equal to a thousandth of the distance segmented along the earth's meridian by one minute of angle.

[4] One *ligne seconde* corresponds to $^1/_{12}$ of a line. See, for example [14].



Estat et proportions des differents corps pour les nouveaux Caractères de l'Imprimerie Royalle

Noms	Grandeur absolue des Caractères Corps	oeil	queüe	Vuide	Lettres Capitalles	Lettres de 2 points	hauteur du poinsson	espaces du Dessus	du Dessous
La petite academie	7. 1/2	2. 13/16	2. 7/64	15/32	4. 9/64	15.	12. 32/64	15/64	2. 15/32
La petite françoise	9.	3. 3/8	2. 13/32	3/16	5. 29/32	18.	14. 29/32	3/32	2. 13/16
La petite dauphine	10. 1/2	3. 13/16	2. 41/64	31/32	6. 17/64	21.	17. 29/64	21/64	3. 3/32
La petite royalle	12.	4. 1/2	3. 3/8	3/4	7. 7/8	24.	19. 7/8	3/8	3. 3/4
Le Louvre	14.	5. 1/4	4. 1/32	13/16	9. 17/32	30.	24. 17/32	15/32	4. 3/16
Le Bignon	18.	6. 3/4	5. 1/16	1. 1/8	11. 13/16	36.	29. 13/16	3/16	5. 1/8
Le Philippeaux	21.	7. 7/8	5. [illegible]	1. [illegible]	[illegible]	42.	34. [illegible]	[illegible]	6. [illegible]
Le Pontchartrain	24.	9.	6. 3/4	1. 1/2	15. 3/4	48.	39. 1/4	3/4	7. [illegible]
Le Louis	30.	11. 1/4	8. 3/8	1. 7/8	19. 11/16	60.	49. 3/8	5/8	9. [illegible]
Le Bourbon	36.	13. 1/2	10. 1/8	2. 1/4	23. 5/8	72.	59. 1/8	1. 1/8	11. [illegible]
Le Grand Louis	42.	15. 3/4	11. 13/16	2. 5/8	27. 3/8	84.	69. 3/8	1. 5/16	13. [illegible]
Le grand Bourbon	48.	18.	13. 1/2	3.	31. 1/2	96.	79. 1/2	1. 1/2	15.
Le grand Louvre	60.	22. 1/2	16. 7/8	3. 3/4	39. 3/8	120.	99. 3/8	1. 7/8	18. 3/4
Le grand Bignon	72.	27.	20. 1/4	4. 1/2	47. 1/4	144.	119. 1/4	2. 1/4	22. 1/2
Le grand Philippeaux	84.	31. 1/2	23. 5/8	5. 1/4	55. 1/8	168.	139. 1/8	2. 5/8	26. 1/4
Le grand Pontchartrain	96.	36.	27.	6.	63.	192.	159.	3.	30.
La grande Academie	120.	45.	33. 3/4	7. 1/2	78. 3/4	240.	198. 3/4	3. 3/4	37. 1/2
La grande françoise	144.	54.	40. 1/2	9.	94. 1/2	288.	238. 1/2	4. 1/2	45.
La grande Dauphine	168.	63.	47. 1/4	10. 1/2	110. 1/4	336.	278. 1/4	5. 1/4	52. 1/2
La grande Royalle	192.	72.	54.	12.	126.	384	318.	6.	60.

Le present estat approuvé pour estre executé selon toutes ses proportions, en gravant les poinçons, frapper et justifier les matrices, fondre les charactères, et [illegible] a l'imprimerie royalle pour servir de regle jusqu'a son entiere execution; fait a versailles par ordre de sa majesté le 13e Juin 1694 signé pontchartrain

Figure 2: Proposed new range of types and metrics, including cap height, x-height, descenders, ornamental initials (in Truchet's hand) — after Jammes [10].

Le Romain du roi, bitmaps, outlines and hints in 1695

The Bignon Commission now turned its attention to existing characters used in printing. Three of its members (Truchet, Jaugeon and Des Billettes) undertook to design "new French letters that we have endeavoured to render as agreeable as possible to the eye".[5] These characters broke with the Garamond tradition. Their originality, explained by Stanley Morison [13] and later by André Jammes [10], lay in the premise that "printing was not a branch of handwriting but a branch of engraving".[6] These three academics produced characters for printing plates (engraved by Simonneau) in the *Romain du roi* font that Grandjean later punched[7] for *Médailles sur les principaux événements du règne de Louis le Grand* in 1702.

Although usually attributed to Jaugeon (the only member of the Bignon Commission with a typographic background), the contribution of a "technician" such as Truchet was without doubt decisive. Indeed, all the manuscripts that have been preserved were written by Truchet (see figure 2). Their study is attributable to André Jammes [10] and James Mosley [14]. While Simonneau's printing plates have been reproduced many times, Truchet's hand-written notes are little known. Yet these are the equivalent of today's AFMs (Font Metrics) and even hinting instructions!

These researchers who lived three hundred years ago were unquestionably geniuses:

- They invented the notion of the "vectorial font" by defining characters in terms of outlines with approximation by arcs of a circle (interpolation by arcs of a circle was still used by Bitstream just ten years ago, before Bézier splines). Characters had already been drawn with a ruler and compass since the 15th century by such people as the Italian Félice Feliciano and later Albrecht Dürer and Tory (Morison [13] published a study on the subject). The innovation came from the use of a fine grid (roughly equivalent to a resolution of 2300 dpi).
- They invented the concept of the bitmap.
- They invented the notion of the "slanted" character (the *Romain du roi* has no italic form; instead, the characters are slanted geometrically by deforming their axes, as seen in figure 4).
- They connected these glyphic specifications with a number of tables or commentaries which are the equivalent of today's AFMs (FM as in Font Metrics), and even hints!

The *Romain du roi* typeface includes all the features of today's digital typefaces — but it is three hundred years old!

Truchet tilings, patterns and ornaments

Just as very few mathematicians, even among those who are concerned with tiling systems or symmetry theory, know the genuine Truchet tilings, typographers are not always aware that ornaments owe much to these patterns.

[5] *Ibid.*

[6] Cf. *The typographic art, two lectures by Stanley Morison*, London, 1949, pp. 27 and 59, quoted by Jammes [10, p. 12].

[7] This character was recently digitised by Frank Jalleau under the name of Grandjean-IN.

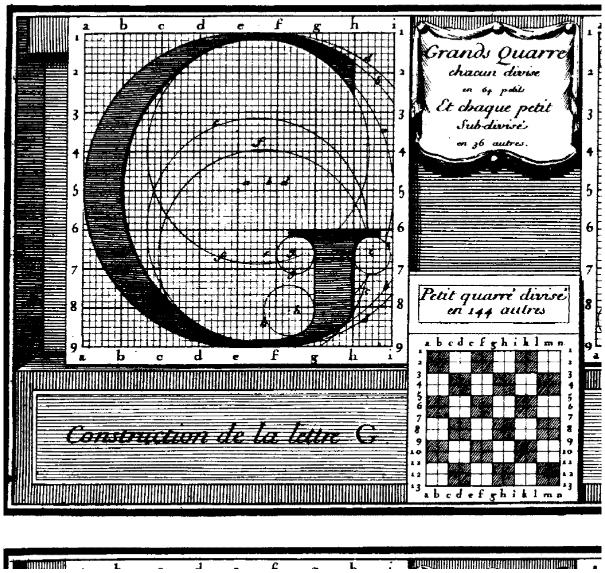

Grands Quarres chacun divisé en 64 petits Et chaque petit Subdivisé en 36 autres.
Petit quarré divisé en 144 autres
Construction de la lettre G

Figure 3: Cartoons of Romain du roi "G", using outlines with circle arcs (design by Truchet; engraving by Simonneau, 1716) — after Jammes [10].

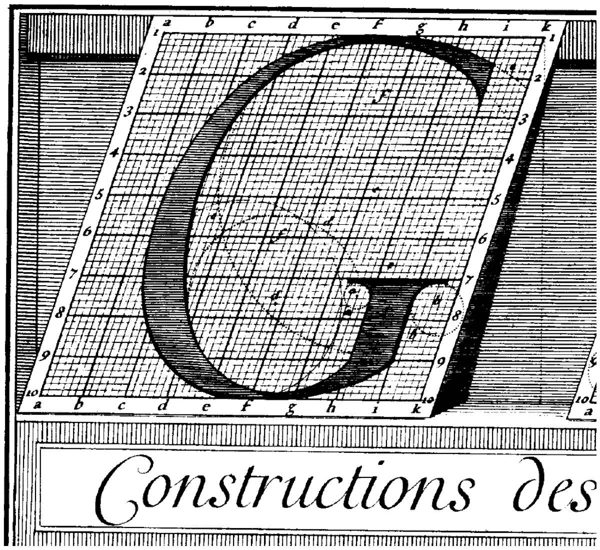


Figure 4: Cartoons of Romain du roi penché (slanted) "G", using geometric deformation of the roman G (design by Truchet; engraving by Simonneau, 1716) — after Jammes [10].

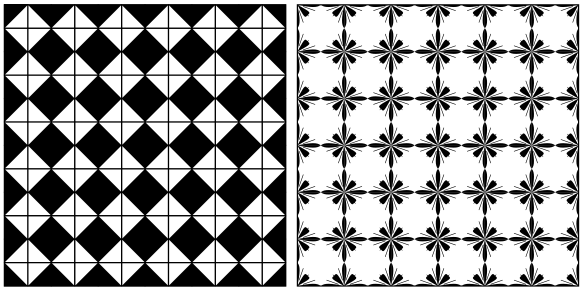


Figure 5: Truchet tiling (left) and a new design using ornaments à la Fournier (right).

At the beginning of the 1700s, Truchet returned to his main field of specialisation: hydraulics. He worked on the Orléans Canal project and wrote, "During the last trip I took to the Canal d'Orléans by order of His Royal Highness, in a château called Motte St Lyé, 4 leagues this side of Orléans, I found several ceramic tiles that were intended for tiling the floor of a chapel and several other apartments. They were of square shape, divided by a diagonal line into two coloured parts. In order to be able to form pleasing designs and patterns by the arrangement of these tiles, I first examined the number of ways in which these tiles could be joined together in pairs, always in checkerboard array." [17]

After some study, in 1704 Truchet published his findings in *Comptes-rendus de l'Académie des Sciences* [18]. This idea of tiling (*pavage*) was not new, but Truchet was the first to publish a systematic study of all combinations of a tiling.

This concept has today become commonplace: if you look closely you will recognise his designs in bathrooms, on kitchen slab floors, in certain decorative elements on monuments and churches, on beehives, and on various articles of clothing. Once again, it was only when mathematicians began to take an interest in the subject — from the middle of the 19th century — that accurate definitions and classifications were established, resulting in a theory. However, the intuitive definition is enough to grasp what the concept of "tiling" entails: composition of a surface by assembling basic parts without blank spaces and without overlapping, using a minimal number of different constituent parts (sometimes just one — think of the tiles in our bathrooms or the cells of honeycombs).

This fundamental principle is nonetheless full of possibilities, especially when several pieces are combined and when they are coloured in a given order. This can produce results that are both visually pleasing and fascinating. This is why from the dawn of time until the present day, and in all civilisations, craftsmen and decorators have made extensive use of these effects. Arab artists, in particular, achieved supreme mastery in this art, only to be equalled in our modern age by the famous engraver Escher [15]. As Islam forbids the depiction of animate objects, the Arab artists and craftsmen turned their hand to producing varied and attractive decorative objects through the sole means of geometry.

While such effects please the eye, it is only because they obey precise mathematical laws, even if these latter were being applied before they'd been discovered! This was a field in which practice took a lead of several centuries over theory. True, Greek scholars had already established a few simple rules on the subject. More importantly, the Greek scholars had grasped certain key principles such as the issue of the discrete or continuous structure of matter, which triggered a famous dispute between Plato and Aristotle known as the "atomism debate". This debate related to the filling of (three-dimensional) space with regular shapes, an extension of the problem of tiling a given (two-dimensional) surface. These questions were barely touched on again until the modern era, except by a few pioneers, most notably the great astronomer Kepler, who devoted fruitful research to it at the beginning of the 17th century. However, the findings of his research were completely forgotten for three hundred years! These early discoverers included Father Truchet, as we have already mentioned.

Only at the end of the 19th century, with the growth of crystallography, did mathematicians take an interest in these issues by applying the theory of groups founded a few decades earlier by Évariste Galois. A major fact that became increasingly apparent at the beginning of the 20th century is that "the geometry of tiling underlies all atomic structures" [11, p. 6]. Throughout the present century, crystallographers and mathematicians, by a kind of process of cross-fertilisation, have advanced the theory of tilings. In the mid 1980s, this theory provided the key to understanding the existence of quasicrystals, which hitherto were thought to be either nonexistent or inconceivable [9, 11, 16].

However, in this rich and complex field many questions, some of which may be written in a single line, remain unanswered. Nevertheless, the findings of tiling theory are used today in a range of different fields: from computer science to image processing, from micro typography to statistical physics, from biology to the creation of lattices for numerical modelling.

Here, too, Father Truchet was a leading, although neglected, forerunner. So let us return to him, or rather, to Fournier: when Truchet was commissioned to work on the Park of Versailles project, his work was continued a few years later by another monk, the Dominican Douat, who published a book (*Méthode pour faire une infinité de desseins différents...* [3]) that Fournier read ten years later.

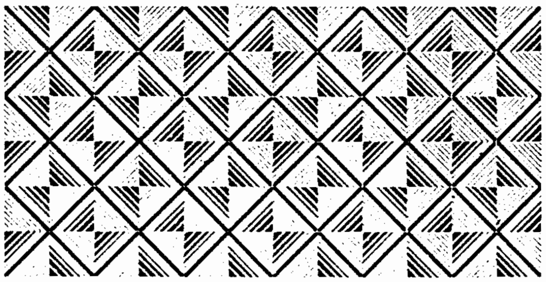


Figure 6: *Tarotée*, based on a Truchet tile (symmetry D) — Fondry Schelter & Giesecke, Leipzig, 1876.

Fournier, originally a wood engraver, developed an interest in ornaments which had hitherto been engraved mainly on wood [4]. Fournier's contribution was three-fold:

- Drawing inspiration from Luce and Douat-Truchet, Fournier divided each element of a large ornament into small elements known as "combinational" ornaments or *vignettes.*
- He made punches from these basic elements (instead of using wood).
- He studied (probably in association with his definition of the point), the strict metric measurements needed to combine these differently sized ornaments.

Apart from a few pages in his *Manuel*, Fournier seems to have made scant use of these ornaments to produce "tiles". On the other hand, Truchet's principle (square tiles divided into two colours by a diagonal line) are present in Luce's ornaments and Fournier's *Manuel* (one third of these 165 ornaments exhibit this symmetry), as well as in the work of Bodoni, Caslon, Peignot, etc.

If some *tarotées*[8] (such as the ones by Jean-François Rosart [1, pages 123–125]), are set with type (instead of using engraving techniques) they are not real Truchet tiles. However, modern *tarotées* (engraved on copper plates) are sometimes real Truchet tiles, like the ones in figure 6 and in specimens from the Laurent and Deberny or Peignot foundries.

Today, fonts offer wood ornaments and fleurons that do not give full justice to this particular art.

[8] Playing cards with grilled or checquered backs. *Tarot* is also the name of a card game: see `http://www.netlink.co.uk/users/pagat/`.

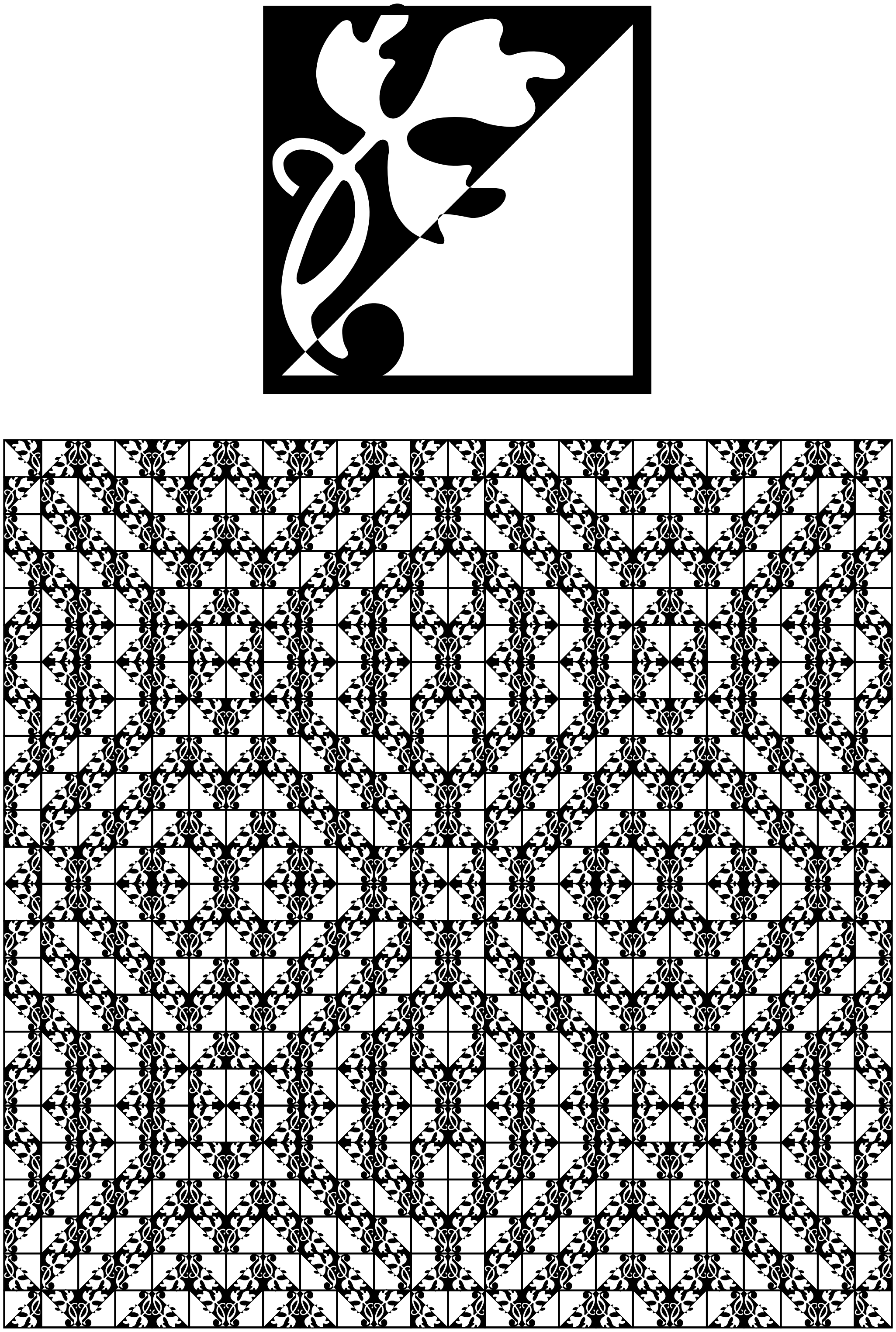


Figure 7: Bottom: a Truchet tiling (n° 64) executed with (at the top) modern ornamentation (Adobe Wood Ornament).

However, from a basic ornament it is easy to create a large number of Truchet tiles by "geometric" composition. These many be easily "programmed" through languages like PostScript or pst-fill [7, 5].

Acknowledgment

The authors wish to express their special thanks to Barbara Beeton and Christina Thiele for their precise, deep, careful ... work when editing this paper.

References

[1] *The type specimen of Jacques-François Rosart, Brussels, 1768.* A facsimile with an introduction and notes by Fernand Baudin and Netty Hoeflake. Amsterdam: Van Gendt & Co., 1973.

[2] Andrew Boag, "Typographic measurement: a sequence of events from Moxon to 1992", *Typography papers*, vol. 1, Reading University, 1996, 105–121.

[3] Dominique Douat, *Methode pour faire une infinité de desseins differents avec des carreaux mipartis de deux couleurs par une ligne diagonale : ou observations du Pere Dominique Douat Religieux Carmes de la Province de Toulouse sur un memoire inséré dans l'Histoire de l'Académie Royale des Sciences de Paris l'année 1704, présenté par le Reverend Sebastien Truchet religieux du même ordre, Academicien honoraire*, imprimé chez Jacques Quillau, Imprimeur Juré de l'Université, Paris, 1722.

[4] John Dreyfus, *Aspects of French Eighteenth Century Typography — A study of type specimens in the Broxbourne Collection at Cambridge University Library.* [Printed for presentation to members of The Roxburghe Club, Cambridge.] Cambridge, University Printing House, 1982.

[5] Philippe Esperet and Denis Girou, "Coloriage du pavage dit 'de Truchet' ", *Cahiers GUTenberg*, num. 31, decembre 1998, 5–18.

[6] Pierre-Fournier dit le jeune, *Manuel typographique utile aux gens de lettres, et à ceux qui exercent les différentes parties de l'Art de l'Imprimerie*, chez l'auteur, Paris, 1764. New English edition: *The Manuel Typographique of Pierre-Simon Fournier le jeune together with Fournier on Typefounding, An English translation of the Text by Harry Carter in facsimile, With an Introduction and Notes by James Mosley*, three volumes; Darmstadt, 1995.

[7] Denis Girou, "pst-fill — a PSTricks package for filling and tiling areas", *TUGboat* 19(2), June 1998, 101–112.

[8] Denis Girou and Jacques André, "Pavages : de Truchet à Escher", *Cahiers GUTenberg*, to appear.

[9] Branko Grünbaum and Geoffrey C. Shepard *Tilings and Patterns*, New York, Freeman and Company, 1987.

[10] André Jammes, *La naissance d'un caractère : le Grandjean — la réforme de la typographie royale sous Louis XIV*, Librairie Paul Jammes (1961) and Promodis, 1985.

[11] Christian Janot and Jean-Marie Dubois, "Les quasicristaux. Matière à paradoxes", *Les éditions de physique*, Paris, 1998.

[12] Edmond Lery, "Le P. Sébastien Truchet, Membre honoraire de l'Académie des Sciences (1657–1749)", *Revue de l'histoire de Versailles et de Seine-et-Oise*, 1929, 220–241.

[13] Stanley Morison, *Letter Forms, Typographic and Scriptorial: Two Essays on Their Classification, History, and Bibliography*, Hartley & Marks Publishers Inc. (2nd printing), 1996.

[14] James Mosley, "French academicians and modern typography: designing new types in the 1690s", *Typography papers*, The University of Reading, 1997, vol. 2, pp. 5–29.

[15] Doris Schattschneider, *Visions of Symmetry: Notebooks, Periodic Drawings, and Related Works of M. C. Escher*, New York, Freeman and Company, 1990.

[16] Marjorie Senechal, *Quasicrystals and Geometry*, New York, Cambridge University Press, rev. ed., 1996.

[17] Cyril S. Smith and Pauline Boucher, "The Tiling Patterns of Sebastien Truchet and the Topology of Structural Hierarchy", *Leonardo*, vol. 20, num. 4, 1987, 373–385.

[18] Sébastien Truchet, "Mémoire sur les combinaisons", *Mémoires de l'Académie Royale des Sciences*, 1704, 363–372. (Some plates and a part of this text appear (in English) in [17].)

◇ Jacques André
Irisa/Inria-Rennes
Campus de Beaulieu
F-35042 Rennes, France
`jandre@irisa.fr`

◇ Denis Girou
Idris/CNRS
BP 167
F-91403 Orsay cedex, France
`Denis.Girou@idris.fr`

Language Support

A Simple Technique for Typesetting Hebrew with Vowel Points

Sivan Toledo

Introduction

This paper describes a simple mechanism for typesetting Hebrew with vowel points. Hebrew uses a large set of accents that represent vowels, consonant modifiers, and cantillation instructions. These accents are placed above, below, or inside letters; a single letter can carry several accents. The solution that we describe, which is designed for PostScript [2] output devices, leaves the placement of the accents to the output device. TEX regards the accents as zero-width characters and does not process them in any special way. Samples of the output are shown in Figures 1 and 2.

The paper only addresses the issue of typesetting vowel and consonant modifiers, known collectively in Hebrew as *nekudot*, or points. We refer to this group of accents as vowels in the rest of the paper. Cantillation marks are only used in printing Biblical texts; they are missing in most fonts, and they complicate the placement of accents. In particular, a cantillation mark attached to one letter may require ajdustments in the positioning of adjacent letters and/or marks attached to them. Vowels, on the other hand, are widely used in Hebrew printing, and are included in most of the Hebrew fonts that have been produced in recent years.[1] A vowel mark attached to a letter does not affect the positioning of other letters or the marks attached to other letters. Without vowels, it is only possible to determine the meaning and pronunciation of many words from their context. Although all proficient Hebrew readers can read most texts without vowels, vowels are still used in printing of liturgical texts, poetry, and texts for beginning readers. Vowels are also used in texts for proficient readers to specify the pronunciation of difficult words.

בְּרֵאשִׁית בָּרָא אֱלֹהִים אֵת הַשָּׁמַיִם וְאֵת הָאָרֶץ:
יִדְרשׁוּן, שֹׂנֵא, וְלַשֹׂרֵקָה, מַצּוֹת, מִצְוֹת

Figure 1: A sample of the output of the vowel-placement mechanism described in this paper. The top line shows the first few words of Genesis. The bottom line shows a few words that are traditionally considered difficult to set [8]. The typeface is a version of Hadassa, designed in the 1950's by Henri Friedlaender.

בְּרֵאשִׁית בָּרָא אֱלֹהִים אֵת הַשָּׁמַיִם וְאֵת הָאָרֶץ
יִדְרשׁוּן, שֹׂנֵא, וְלַשֹׂרֵקָה, מַצּוֹת, מִצְוֹת

Figure 2: Another example; the font is Omega Serif Hebrew (a version of a typeface called David, designed by Ismar David in the 1950's).

The technique described in this paper is designed for typesetting Hebrew with vowel marks. The placement of a vowel mark is determined by the specific letter it is attached to and possibly by other marks attached to the same letter. The technique does not allow the positioning of marks or letter to depend on adjacent letters and their marks, so it may be insufficient for setting Hebrew with both vowels and cantillation marks.

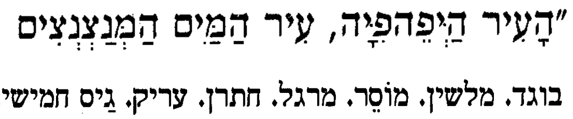
Two factors contribute to the difficulty of typesetting Hebrew with vowels. First, the large number of possible letter-vowel combinations makes it impractical to use a separate glyph for each possible combination. Second, every vowel needs to be placed at a particular location relative to each letter (one can decompose the set of vowels into classes of vowels with the same placement, but there are at least 7 such classes). Generally speaking, a well chosen placement visually centers the vowel with respect to the letter, but there are exceptions.

Because of these technical difficulties, one often finds texts in which the vowels are placed incorrectly. In some cases, the placement is just off center, but in other cases it is completely incorrect, to the extent that the vowel appears on the wrong side of the letter or under the wrong letter. This happens even in Bibles (see [8] for examples) and in books that were recently published by major Israeli publishers, as shown in the samples in Figure 3.

This paper does not explain how vowel points should be set; for information on this topic, see [5, 6, 8, 11, 14]. For further information on Hebrew typography in English, see [12]; this article is a bit dated, but is widely available in libraries. The

This work was done while the author was at the Xerox Palo Alto Research Center in Palo Alto, California.

[1] Nearly all the TrueType fonts that come with the Apple Macintosh's Hebrew Language Kit (`www.apple.com`), with Microsoft's Hebrew Windows, and with Hebrew word processors include vowel points (Microsoft's Hebrew products are available outside Israel from Glyph Systems, `www.glyphsys.com`). So do nearly all the commercial PostScript and TrueType fonts that are produced by Elsner and Flake (`fontinform@t-online.de`; distributed in Israel by Panergy, `www.inter.net.il/~panergy`), by Studio Rosenberg in Israel (with the exception of their display typefaces; `mailto:master_f@netvision.ac.il`), and by Monotype (`www.monotype.com`).



”הָעִיר הַיְפֵהפִיָּה, עִיר הַמַּיִם הַמְנַצְנְצִים

בוגד. מלשין. מוֹסֵר. מרגל. חתרן. עריק. גַיס חמישי.

Figure 3: Examples of the poor setting of vowel point in books that were recently published by major Israeli publishers. The top sample is a true typesetting disaster, with totally incorrect placement of many vowels. The bottom sample is generally well set, except for the leftmost vowel sign which is set too much to the right.

collection [13] contains more up-to-date information in Hebrew.

An Overview of the Vowel Placement Mechanism

This section describes the mechanism that I have developed for placing Hebrew vowels.

The main idea behind the mechanism is to leave the precise placement of the vowels to the PostScript font program that the output device uses, rather than require the main typesetting engine, TEX, to place these accents. To TEX, vowels appear to be regular characters with zero widths. The advantages and disadvantages of this technique over other possible vowel-placement mechanisms are discussed in Section Discussion and Comparison to Related Techniques.

In the source text file, the vowels are placed after the base letter, in accordance with Unicode usage [14] (and unlike the current text-entry mechanism for Latin accents in TEX). When TEX--XET (a bidirectional version of TEX) processes the file, the accents are placed to the left of the base letter. Once the dvi file is processed by a dvi-to-PostScript processor, they are placed in the PostScript output file *before* the base letter, since the order of the characters of a word within right-to-left text is reversed.

The PostScript font that the Hebrew is set in is responsible for vowel placement. We use a Type 3 font [2], in which the rendering of each glyph causes the invocation of an arbitrary PostScript procedure. When this procedure is asked to render a vowel mark, it does nothing, except *remember* (raise a boolean flag) that this vowel must be set when the next base letter is to be rendered. When the procedure is asked to render a base letter, it renders the base letter and then renders all the vowel points whose flags are raised, and clears the flags. Since the rendering of the base letter and vowel points is now done together in the same PostScript procedure, we can easily control the placement of the vowels with respect to the letter.

This technique is insensitive to the ordering in the input file of the vowels that apply to a single base letter, which is consistent with the Unicode specification.

The Type 3 font renders the letters and vowel points by calling another font program, not by specifying the actual shapes of the glyphs. The Type 3 font therefore behaves in much the same way that a virtual TEX font behaves, but it is considerably more flexible, since it can use all the facilities of the PostScript language. In particular, current virtual TEX files are limited to 256 glyphs, which prevents them from being used to specify all the possible letter-vowel combinations in Hebrew.

There are several advantages to calling another font to render the glyph rather than specifying their shapes in the Type 3 font. First, since we use unmodified Type 1 [1] and TrueType [9] fonts (encapsulated in a Type 42 PostScript font [3]), we preserve their hints and avoid license violations that may be associated with a conversion of the outlines to a Type 3 font that is subsequently modified. We also gain in PostScript rasterization speed, since the glyphs of the Type 3 must not be cached (the rasterizer must call the font to "render" vowel points, and the shape of base letters depends on their vowels). But the glyphs of the Type 1 or Type 42 font that the Type 3 font calls are cached. Still, the placement of vowels by the Type 3 font slows down rasterization considerably compared to the rasterization of a text set in a Type 1 font, but this problem can be alleviated by optimizing the procedures of this Type 3 font, or by conversion to PDF format [4], as explained below. Early experiments with a Type 3 font that both placed vowels and specified the outlines of glyphs revealed that when PostScript files that use such fonts are distilled to PDF using Adobe Distiller, the resulting PDF file encodes the outline of every glyph that appears on the page separately. That is, in a document that used 100 different glyphs 100,000 times, the PDF file would include 100,000 glyph outlines, not just 100. This produces a large file that renders slowly. In contrast, when the Type 3 font calls a Type 1 font, the PDF file encodes only the placement of every glyph on the page, and rendering speed is essentially the same as the speed of rendering a page set in a Type 1 font.

Technical Details

This section describes the implementation of the Type 3 font that places vowels.

The BuildChar and BuildGlyph procedures of the font are fairly standard: the BuildChar procedure calls BuildGlyph, and BuildGlyph calls the appropriate procedure from a CharProcs dictionary, after pushing on the stack the CharStrings dictionary itself. It also pushes a mark, so that the stack can be cleaned up later.

```
/BuildGlyph {
  mark
  3 1 roll
  exch /CharProcs get
  dup
  3 -1 roll
  2 copy known not {pop /.notdef} if
  get exec
  cleartomark
} bind def

/BuildChar {
  1 index /Encoding get exch get
  1 index /BuildGlyph get exec
} bind def
```

The CharProcs dictionary includes procedures that render base letters, that “render” vowels (in effect, these just raise a flag), and procedures that perform the actual rendering of vowels based on the state of the flags. The dictionary also includes the flags themselves.

```
/CharProcs 200 dict def
CharProcs begin

% vowel flags
/sheva_flag       false def
/hatafsegol_flag  false def
...
```

When the PostScript rendering engine needs to render a vowel, it calls a procedure similar to the one below, which simply sets the appropriate flag. The font metrics file describing this font must reflect, of course, the fact that the width of the vowels is 0. (It does not matter what the width of the vowels in the font that is used for actual rendering is.)

```
/sheva {
  % zero-width character, do not cache
  0 0 setcharwidth
  % set the flag
  /sheva_flag true put
} def
```

When the PostScript rendering engine needs to render a base letter, it calls a procedure like the next one. The procedure first renders the letter by calling another font, and then calls procedures that render vowels, if the appropriate flags are set. The horizontal and vertical offsets for each vowel are specified separately. The reason that the width of the base letter is specified as 1/1000 the width specified in the afm metrics file, and that the scaling of the font that renders the glyph is 1, is that we specify the font transformation matrix (the font’s `/FontMatrix`) for this Type 3 font as [1 0 0 1 0 0].

```
/bet {
  % width taken from .afm file
  0.614 0 setcharwidth
  % render using another font
  0 0 moveto
  /MonotypeHadassah findfont
  1 scalefont setfont
  (\341) show
  % now set placement for each vowel and call
  % /draw... to render it if necessary.
  % The sheva, for example, is offset 0.250
  % units to the right with respect the a bet.
  dup 0.250 0.000 4 -1 roll /drawsheva get exec
  dup 0.300 0.000 4 -1 roll /drawhiriq get exec
  ...
  dup 0.220 0.000 4 -1 roll /drawtsere get exec
  % this vowel, holam, is moved both
  % horizontally and vertically
  dup -0.04 -0.08 4 -1 roll /drawholam get exec
} def
```

The next set of procedures includes the ones that test the flags, clear them, and render the glyphs when necessary.

```
/drawsheva {
  % arguments are already on the stack
  moveto
  % get the flag, leave on stack
  dup /sheva_flag get exch
  % clear the flag
  dup /sheva_flag false put exch
  % if set, render
  {dup /sheva_glyph get exec}
  if
} def
...

/sheva_glyph {
  /MonotypeHadassah findfont
  1 scalefont
  setfont (\300) show
} def
...
```

Discussion and Comparison to Related Techniques

The vowel-placement technique described in this paper allows for accurate placement of vowels of Hebrew. It was designed for use with TEX--XET, but it can also be used with other typesetters that produce similar PostScript output. It can be used with any unmodified Type 1 or TrueType font (encapsulated in a Type 42 font). The technique requires only one software component besides TEX--XET and a Hebrew PostScript font: a Type 3 PostScript font that places vowels. The end user simply types the Hebrew text with vowels following the base letter (as prescribed by Unicode) and runs the file through TEX--XET and a dvi-to-PostScript processor. The resulting output file can be converted to a high-quality PDF file that renders quickly. Since the technique does not use any TEX macros, it does not interfere with any, and texts with vowels can be freely used in moving arguments, indices, etc.

Producing a new Type 3 font that specifies vowel placement takes a few hours, even without interactive visual tools (I used a text editor and a PostScript previewer). The production process can probably be sped up using an interactive placement editor, or at least a table-driven script that would generate the actual font. Producing such fonts requires no special TEX expertise. The Type 3 font that places vowels for the Omega Serif Hebrew font is freely available from the author.

The technique has other potential advantages, which I have not yet exploited, but are worth mentioning. It is possible to choose narrow vowel glyphs for narrow letters, thus ensuring that the vowel does not extend beyong the letter. It is possible to render the vowels in a different color than letters; I think that this may be useful in large sizes, where rendering the vowels in gray, so as to emphasize the base letters.

The main disadvantages of this technique are that the resulting output files render somewhat slowly and that it is fragile. The PostScript rasterization speed is slow because the PostScript interpreter computes glyph placements algorithmically. As explained above, the problem can be alleviated by optimizing the Type 3 fonts or by conversion to PDF, especially for documents that are designed to be read on a computer screen. The method is fragile because it assumes that TEX and the dvi processor always place the vowels before the base letter in the PostScript file, and that the PostScript interpreter would render the glyphs in the order in which they appear in the file. These assumptions are true today, but they may change in the future. Still, I have not encountered any robustness problems yet.

Since letter-vowel combinations are rendered by a font program, they must be rendered with no information about nearby glyphs. For example, it is not possible to render vowels based on the vowels of the letters to the right and left of the current one, or based on whether the letter appears at the beginning or end of a line. This is not a serious limitation for typesetting Hebrew with vowel marks, but it a limitation when typesetting biblical texts with both vowel and cantillation marks.

There are other solutions and partial solutions to the vowel-placement problem. One common partial solution is to use fonts that have some precombined letter-vowel glyphs. A set of around 50 or 60 precombined glyphs, which are treated by the typesetting software as ligatures, ensure an accurate placement of the most difficult combinations. The other combinations are produced by overstriking a base letter or one of the precombined glyphs with vowel glyphs. Overstriking usually places the vowel slightly off center (because a single overstriking vowel glyph is used for all base letters). The typographical quality of the result is not high, but it is not a disaster either. Also, to ensure that ligatures are used, the user must type vowels in a specific order, which is inconsistent with Unicode usage. For example, if the font includes a ligature for the letter yod with a dagesh mark, the user must type the dagesh before other vowels that apply to the yod. This type of solution appears to be fairly common in commercial publishing in Israel.

Another solution, specifically for TEX, was developed by Haralambous [5, 6]. His system, which sets both vowel and cantillation marks, uses a preprocessor and a special font to place these accents. I do not know whether the system can be adapted to other fonts. While Haralambous's system is more robust and flexible than the one I describe, my approach is simpler to implement and use.

Haralambous is also one of the developers of Omega [7] (with Plaice), a system for multilingual typesetting that is based on TEX, uses a 16-bit character set (Unicode), adds to TEX another processing mechanism that is separate from macros, and has more flexible virtual fonts. Omega should enable typesetting Hebrew with vowel points.

The introduction of OpenType [10] fonts will probably encourage the development of additional solutions. OpenType fonts, which were developed by Adobe and Microsoft, are essentially enhanced TrueType fonts that can contain Type 1 outlines and advanced typographical layout information, similar in spirit to the information that Apple's TrueTypeGX fonts can contain. OpenType fonts are already used in Arabic and Far Eastern versions of Microsoft Windows, and they are supposed to be used in future versions of all versions of Windows, including Hebrew. It is possible that Hebrew OpenType fonts with enough information for accurate placement of vowel points will be produced. There are currently no such fonts, however, and no programs that can use them. Converting OpenType fonts and the typographic information that they contain to a format that TEX can use is certainly a complex task.

All in all, I found this technique to be a remarkably simple solution to a complex problem.

Acknowledgements

Thanks to Barbara Beeton, Yannis Haralambous, and an anonymous referee for helpful comments.

References

[1] Adobe Systems Incorporated. *Adobe Type 1 Font Format*. Addison Wesley, 1990.

[2] Adobe Systems Incorporated. *PostScript Language Reference Manual, 2nd Edition*. Addison Wesley, 1990.

[3] Adobe Systems Incorporated. *The Type 42 Font Format Specification*. Technical Note 5012, 1993.

[4] Tim Bienz, Richard Cohn, and James R. Meehan. *Portable Document Format Reference Manual, Version 1.2*. Adobe Systems Incorporated, 1996.

[5] Yannis Haralambous. Typesetting the Holy Bible in Hebrew, with TEX. *TUGboat*, 15(3): 174–191, 1994. Also appeared in the *Proceedings of EuroTEX 1994*, Gdańsk, 1994.

[6] Yannis Haralambous. “Tiqwah”: a typesetting system for biblical Hebrew, based on TEX. *Bible et Informatique*, 4:445–470, 1995.

[7] Yannis Haralambous and John Plaice. Omega, a TEX extension including Unicode and featuring lex-like filtering processes. *Proceedings of EuroTEX 1994*, Gdańsk, 1994.

[8] Eliyahu Koren. [The letter as an element in the design of sacred books]. In Hebrew. In [13], pages 85–90.

אליהו קורן. האות כיסוד בעיצוב ספרי קודש.

[9] Microsoft Corporation. *TrueType Specification, Version 1.66*. Available online at `www.microsoft.com/typography`.

[10] Microsoft Corporation. *OpenType Specification, Version 1.1*. Available online at `www.microsoft.com/typography`.

[11] Gershon Silberberg with contributions by Moshe Spitzer, Meir Ben-Yehuda, Shmuel Perez, and Arie Lotan. Principles of Printing. Irgun Mifale ha-Defus be-Yisrael, Tel Aviv, 1968.

גרשון סילברברג, בהשתתפות משה שפיצר, מאיר בן־יהודה, שמואל פרץ, ואריה לוטן. **תורת הדפוס**. ארגון מפעלי הדפוס בישראל, תל־אביב, תשכ״ב.

[12] Moshe Spitzer. Typography. *Encyclopedia Judaica*, 15:1480–1488, Keter, Jerusalem, 1971. In English.

[13] Moshe Spitzer, editor. *[A Letter is Forever: A Collection of Papers on the Design of the Hebrew Letter]*. In Hebrew. Second edition, Israel Ministry of Education and Culture, 1989 or 1990.

משה שפיצר, עורך. **אות היא לעולם: קובץ מאמרים מוקדש לעיצוב האות העברית.** משרד החינוך והתרבות, האגף לחינוך תורני, ירושלים, תש״ן.

[14] The Unicode Consortium. *The Unicode Standard Version 2.0*. Published by Addison-Wesley, 1996. Parts of the standard are available online at `www.unicode.org`.

◇ Sivan Toledo
School of Mathematical Sciences
Tel-Aviv University
Tel-Aviv 69978
ISRAEL
sivan@math.tau.ac.il

TEX Live CD-ROM

An introduction to TEX Live 4

Editor: Sebastian Rahtz
sebastian.rahtz@oucs.ox.ac.uk

1 Introduction

This article[1] is rewritten from that published in *TUGboat* for TEX Live 3 in 1998 and describes the main features of the TEX Live 4 CD-ROM—a TEX/LATEX distribution for Unix, Linux, Windows32 (and other) systems, and a wide-ranging set of macros, fonts and documentation conforming to the *TEX Directory Standard* (TDS)—which can be used with nearly every TEX setup.

The CD-ROM bundled with this issue of *TUGboat* is being provided as a benefit of 1999 TUG membership. To keep up-to-date on the the TEX Live project, please visit its Web page.[2] The TEX Users Group recognizes the importance of the TEX Live CD-ROM and supports its development and production. Volunteers to assist with this work are encouraged to contact tex-live@tug.org.

A fuller version of this document (in English, French, German and Slovak) can be found on the CD-ROM in tldoc.

1.1 Changes since TEX Live 3

Although there have been no structural changes, a very great many changes have been made, some more visible than others. Changes that users should know about include:

1. The main TEX programs are based on Web2c version 7.3;
2. Both Unix and Win32 versions are identical to teTEX 0.9 (as of the end of March 1999), and simply add more programs, and a much larger support tree;
3. New programs include dvipdfm (DVI to PDF driver) and tth (TEX to HTML converter), as well as new versions of pdfTEX, Ω, and ε-TEX;
4. The 'December 1998' (actually March 1999) LATEX is included;
5. A brand new Windows install program is provided;
6. A great many font and macro packages have been updated;
7. Packages are now starting to be classified as 'free' or 'non-free' (according to the Debian Free Software Guidelines[3]) and we expect during the coming year to complete this work, and be able to offer a genuinely 'free' TEX CD-ROM at the start of 2000.

Naturally, much effort has been expended on testing the structural integrity of the texmf tree (in particular, checking it against what teTEX does, and checking the licensing conditions of packages).

We very strongly urge any package authors reading this to consider looking at how their work

[1] The guide to kpathsea and web2c was written by Michel Goossens, and Fabrice Popineau wrote the section on Windows installation and use. The TEX Live CD-ROM distribution is a joint effort by the TEX Users Group, and the UK, French, German, Czech/Slovak, Dutch, Indian and Polish user groups. For the 1999 edition, we are particularly grateful to:

- Karl Berry, who provided the original Web2c distribution, and has continued to give invaluable advice, encouragement, and help;
- Mimi Burbank, who arranged access at the Florida State University Supercomputer Research Institute to a slew of different computers to compile TEX on, and acted as an essential guinea-pig whenever asked;
- Kaja Christiansen, who provided essential feedback and documentation preparation;
- Thomas Esser, without whose marvelous teTEX package this CD-ROM would certainly not exist, and whose continual help makes it a better product;
- Eitan Gurari, whose TEX4ht was used to create the HTML version of this documentation, and who worked tirelessly to improve it at short notice;
- Art Ogawa and Pat Monohon, who coordinated this release for TUG;
- Petr Olšák, who coordinated and checked all the Czech/Slovak material very carefully;
- Fabrice Popineau, who has has worked away unceasingly at the Win32 part of the package (especially the setup!) and contributed in many different ways with ideas, advice and code;
- Walter Schmidt, who checked the emTEX and OS/2 material, and found other horrors;
- Staszek Wawrykiewicz, who provided great checking feedback, and coordinated the Polish contributions;
- Olaf Weber, for his patient assembly and maintenance of Web2c 7.3;
- Graham Williams, on whose work the catalogue of packages depends.

Alain Rabaute, Pascal Quignon, Gerhard Wilhelms, Fabrice Popineau, Libor Skarvada, Staszek Wawrykiewicz, Erik Frambach, and Ulrik Vieth kindly translated documentation into their respective languages, checked other documentation, and provided very welcome feedback.

[2] http://www.tug.org/tex-live.html

[3] http://www.debian.org/intro/free

is arranged on the CD-ROM, and contacting us with any problem. But most importantly, look at the *Catalogue* maintained by Graham Williams[4] and check your details in there, especially the licensing! Future versions of TEX Live will rely more and more on the *Catalogue*.

2 Structure and contents of the CD-ROM

The important CD-ROM top-level directories are listed below.

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

tldoc Documentation for TEX Live;

FAQ Frequently Asked Questions, in English, French, and German;

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

man Documentation in the form of Unix man pages for the TEX system;

source The source of all programs, including the main Web2c, TEX, and METAFONT distributions – stored in a compressed tar archive;

support Various bits of TEX-related software which are *not* installed by default, such as MusixTEX, support programs, and a complete distribution of Ghostscript version 5.50;

systems Packaged TEX systems which are separate from the main TEX Live. Subdirectories in here are:

- **macintosh** The CMacTeX package ready to install;
- **msdos** The emTEX package for MS-Dos;
- **os2** The OS/2 TEX package emTEX/TDS and the EPMTFE TEX shell for the EPM editor.

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

usergrps Material about TEX User Groups.

2.1 The TDS tree

The TEX Live `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. 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.

etex Support for ε-TEX.

fonts Font sources, metrics, PostScript and bitmap forms.

formats Eplain, RevTEX, phyzzx, 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.

omega Support for Ω.

pdftex Support for pdfTEX

plain Macros for plain TEX.

systems Binaries for Unix and Win32 platforms.

texlive Basic material for the distribution.

Each of the collections is divided into *basic* (1), *recommended* (2), and *other* (3). 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 `texmf/lists` contains lists of all files in each package (used by the installation programs). Graham William's *Catalogue* lists all the packages, noting whether they are in TEXLive, and giving details.

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.

[4] `http://www.cmis.csiro.au/Graham.Williams/TeX/catalogue.html`

Warning: This CD-ROM is in ISO 9660 (High Sierra) format, with Rock Ridge and Joliet 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 this with the others.

3.1 Running from the CD-ROM under Unix

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 shows the list of available systems and the corresponding directories.

DEC Alpha OSF/1	`alpha-osf4.0`
HP9000 HPUX	`hppa11-hpux10.10`
Intel Linux	`i386-linux` `i386-linux-libc5`
SGI IRIX	`mips-irix6.2`
IBM RS 6000 AIX	`rs6000-aix4.1.4`
Sun Sparc Solaris	`sparc-solaris2.5.1`
Windows 95 or NT (Intel)	`win32`

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 `texmf-localconfig` on the CD-ROM (which does not exist!), so you *must* override this by setting the `VARTEXMF` environment variable.

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

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

Then they should include the directory containing the binaries for the given architecture into the search path by updating the `PATH` variable.

```
PATH=/cdrom/bin/i386-linux:$PATH
export PATH
VARTEXMF=/usr/TeX.local
export VARTEXMF
```

For convenience, these statements can also be entered into the `.profile` script.

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 initialise things, and check it all works.

3.2 Installing 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 `bash`.) This script 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 initializing.
Counting selected collections... Done counting.
Calculating disk space requirements for collections...
          Done calculating that.
Initializing system packages...
          Done initializing system.
```

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

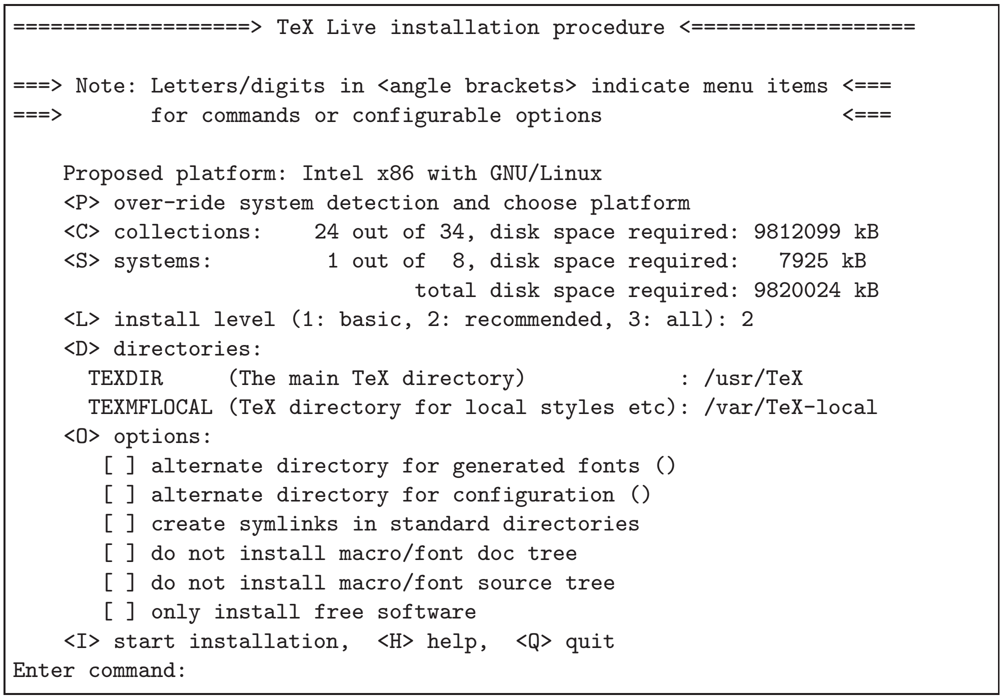
1. the type of system you are on, or want to install for;
2. the collections you want to install, at the *basic*, *recommended* or *other* level;
3. the location on your hard disk to put the files;
4. some runtime behaviour features.

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/TeX`; note that the disk space required for the current installation configuration is also displayed. If you make a suggested setup, you need about 100 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 `texmf`.

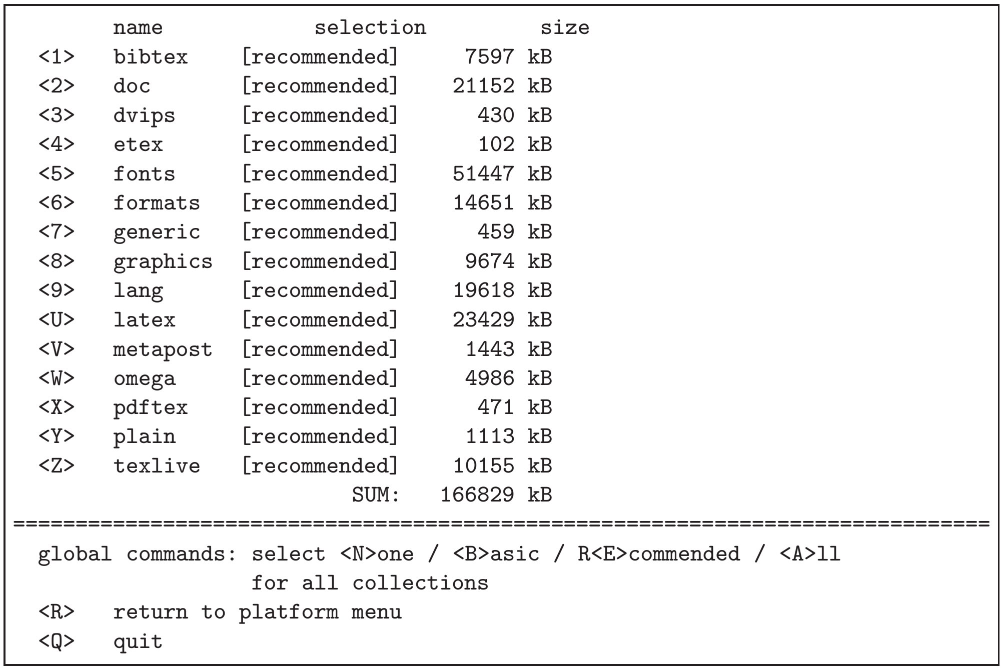
The `options` item lets you decide whether to make new fonts be created in another location (if you want the main package mounted read-only for most users), and whether to make symbolic links for the `man` and GNU `info` pages in the 'standard' locations; you'll need 'root' permissions for tasks to do this, of course.

When you choose `<C>` for 'collections', you will see the display of available collections, the level of



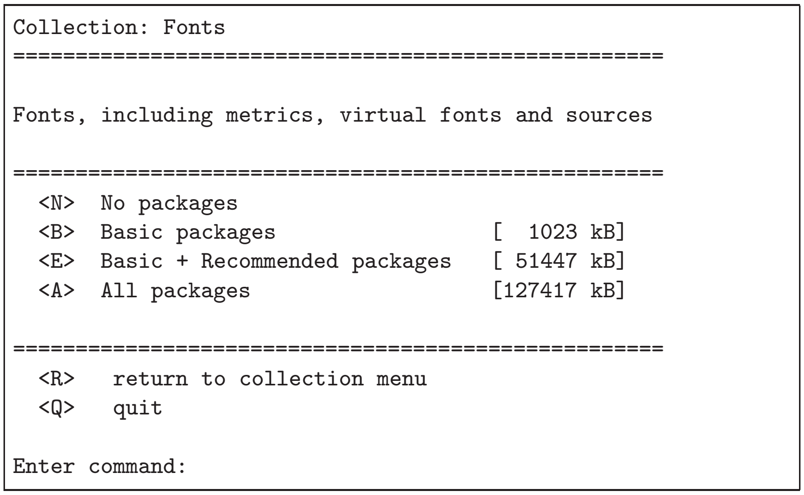
```
===================> TeX Live installation procedure <==================

===> Note: Letters/digits in <angle brackets> indicate menu items <===
===>       for commands or configurable options                   <===

    Proposed platform: Intel x86 with GNU/Linux
    <P> over-ride system detection and choose platform
    <C> collections:    24 out of 34, disk space required: 9812099 kB
    <S> systems:         1 out of  8, disk space required:    7925 kB
                                total disk space required: 9820024 kB
    <L> install level (1: basic, 2: recommended, 3: all): 2
    <D> directories:
      TEXDIR     (The main TeX directory)            : /usr/TeX
      TEXMFLOCAL (TeX directory for local styles etc): /var/TeX-local
    <O> options:
       [ ] alternate directory for generated fonts ()
       [ ] alternate directory for configuration ()
       [ ] create symlinks in standard directories
       [ ] do not install macro/font doc tree
       [ ] do not install macro/font source tree
       [ ] only install free software
    <I> start installation,  <H> help,  <Q> quit
Enter command:
```

Figure 1: Main control screen



```
        name           selection         size
  <1>   bibtex    [recommended]     7597 kB
  <2>   doc       [recommended]    21152 kB
  <3>   dvips     [recommended]      430 kB
  <4>   etex      [recommended]      102 kB
  <5>   fonts     [recommended]    51447 kB
  <6>   formats   [recommended]    14651 kB
  <7>   generic   [recommended]      459 kB
  <8>   graphics  [recommended]     9674 kB
  <9>   lang      [recommended]    19618 kB
  <U>   latex     [recommended]    23429 kB
  <V>   metapost  [recommended]     1443 kB
  <W>   omega     [recommended]     4986 kB
  <X>   pdftex    [recommended]      471 kB
  <Y>   plain     [recommended]     1113 kB
  <Z>   texlive   [recommended]    10155 kB
                           SUM:   166829 kB
==============================================================================
  global commands: select <N>one / <B>asic / R<E>commended / <A>ll
                   for all collections
  <R>   return to platform menu
  <Q>   quit
```

Figure 2: Selecting collections



```
Collection: Fonts
====================================================

Fonts, including metrics, virtual fonts and sources

====================================================
  <N>  No packages
  <B>  Basic packages                [  1023 kB]
  <E>  Basic + Recommended packages  [ 51447 kB]
  <A>  All packages                  [127417 kB]

====================================================
  <R>   return to collection menu
  <Q>   quit

Enter command:
```

Figure 3: Customizing a collection

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. If you installed a system, an initialisation sequence is now 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, you can move the binaries up one level, e.g., from `/usr/local/bin/alpha-osf3.2` to `/usr/local/bin`; if you do this, however, you must edit `texmf/web2c/texmf.cnf` (see Appendix 7) and change the line near the start which reads

```
TEXMFMAIN = $SELFAUTOPARENT
```

to

```
TEXMFMAIN = $SELFAUTODIR
```

If you move the whole installation to another directory tree entirely, you need to edit `TEXMFMAIN` to specify the support tree explicitly, and set `TEXMFCNF` in your environment to `$TEXMFMAIN/texmf/web2c`.

3.3 Installing individual packages 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 `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 `mktexlsr` 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 it, 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/free/latex3/fancyhdr
texmf/source/latex/fancyhdr/README
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 `natbib`:

```
>> sh install-pkg.sh --package=natbib
```

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

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

- Install all the packages available in the *other* Plain TeX collection:

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

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

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

3.4 The texconfig program

After the installation program has copied all files to their final locations, you can use 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, included as part of the binary packages) or command-line interface. It should be used for all maintenance, such as changes of installed printers, or rebuilding the file database. Both modes have help text to guide you through the facilities.

4 Installation and use under Windows

This section only applies to systems running Windows 9x or NT. If you run Windows 3.1, you will have to install `emtex` from the top level `systems` directory by hand.

It is also necessary to have your Windows set up so that it uses the Microsoft Joliet extensions for reading CD-ROMs; simply look at the CD-ROM in Windows Explorer and see whether it shows long, mixed-case file names. If it does not, you cannot use the ready-to-run system on the CD-ROM.

4.1 What is fpTeX?

The system on the CD-ROM for Windows is Fabrice Popineau's fpTeX. This is a port to Windows 9x and Windows NT– referred to as Win32 – of the well known distribution teTeX for Unix. More precisely, given obvious differences between Unix and Win32, some things behave differently under fpTeX, some are still missing, some are just different, but the large majority behave just the same as under Unix.

4.2 What is in this port

This version includes all the same programs as the Unix side of TeX Live, with some additions which can be installed.

Among the programs, there are a few DLLs:

- the ones that begin with msvc are the Microsoft C library targeted for multi-threaded applications,
- the Kpathsea dynamic-linked library,
- `zlib.dll` (compress library) and `libpng.dll` (Portable Network Graphics) for pdfTeX and a few other programs,
- `tex.dll`, `pdftex.dll` and a few others refer to TeX engines – see the explanation below.

All the various TeX engines are distributed in the form of a `.dll` file for the core, and a `.exe` file for the front-end. This is the answer to the problem of link files that do not exist on Win32 platforms. For example, the TeX engine is made up of:

```
11/19/98  11:07a      217,088 tex.dll
11/19/98  11:07a       16,384 tex.exe
```

and the `latex.exe` file is nothing but a rough copy of `tex.exe` using the same core `tex.dll`. The same approach is used for the `mktex*.exe` family of programs which are linked to the `mktex.dll` library.

4.3 Running from the CD-ROM

You can run all the TeX programs directly off the CD-ROM, and have access to all the macros and fonts immediately, at the price of a slower performance than if you install on the hard disk. To do this, you must add the `bin/win32` directory of the CD-ROM to your `PATH`, using the Windows configuration software. Now you can run the programs at a command prompt, or use the shareware WinEdt editor, which runs the programs from convenient menus.

4.4 How to install it

When you put TeX Live in your computer, the setup should run automatically; if it does not, run the program `autorun.exe`. Now follow the instructions—here are some hints:

- Choose a *root* for your installation, `c:\TeX` is proposed by default, but you can change it because you will need lots of disk space: more than 300Mb for a full installation, and beware that the cluster size on FAT partitions will make the package appear even bigger;
- Do not use any path with embedded 'space' character: TEX won't like that. The `setup.exe` checks for that anyway;
- Your *main* `texmf` tree will be *root*/`texmf` and designated by the variable `$TEXMFMAIN`;
- You have the opportunity to add some more `texmf` trees:
  - one *local* `texmf` tree, which is designated by the variable `$TEXMFLOCAL` and is by default *root*/`texmf.local`. It is intended to store your site local macros and style files, and also any locally generated font files. *If you do not specify a local `texmf` tree, you will be asked to set the variable `VARTEXFONTS` to point to some directory where those fonts will be stored. If you specify a local `texmf` tree, it will be used for those fonts.*
  - one *home* `texmf` tree, which is designated by the variable `$HOMETEXMF` and is by default `$HOME/texmf`. This is meaningful only under Windows NT, where users have a `$HOME`. Usually, Windows 9x users do not have a `$HOME`, so they should leave this place empty;

  These locations can be edited manually by looking for their variables names in the file `texmf/web2c/texmf.cnf`.
- You will be asked if you want to install:
  - PK fonts,
  - only *free* packages, in which case some packages will even disappear from the list for a *custom* installation,
  - if you want the documentation accompanying each of the packages being installed; guides and general documentation will always be installed, but not the material that is provided with some specific package;
  - *idem* for source files.
- Choose your setup from *basic*, *recommended* and *full*. If you already know what you want to install, you can also choose to do a *custom* setup. In this case, you will be presented with a list of collections of packages. Each collection is (de)selectable by itself, and so is each package individually. A short description is provided for many packages, thanks to the Web catalogue of Graham Williams.
- Ignore the 'setup' collection in the list of collections. It refers to the files in the `\setupw32` on the CD-ROM and should have been made invisible;
- You will have the opportunity to add some more special packages, namely Ghostscript and Ghostview, the Postscript interpreter and previewer, ImageMagick which allows for image manipulations and conversions, WinEdt which is a good shareware environment for editing and typesetting with TEX, and texshell which does quite the same as WinEdt (free but less fancy).
- You can review your installation settings, and if everything is okay, the file transfer will begin;
- Once it is done, the installers for Ghostview and WinEdt will be called if selected. The installation and configuration of everything else is handled by `setup.exe`. It is completely automatic. The last step is the build of 'ls-R' files;
- Eventually, the setup will be over and the documentation displayed using your default Web browser. Windows 9x users will need to reboot before being able to run anything.

A number of items will appear under the `Start->Programs->TeXLive` menu.

4.5 What does the setup hide from me ?

If you want to hack the configuration by yourself, here is a more detailed description of what the `setup` does — and what it does not. Your `PATH` is modified to make the programs installed accessible. It is checked for any older version of fpTEX or TEX-Live, and if one is found, its entry in `PATH` is removed. This is done by looking for the file `kpathsea.dll` along your `PATH`.

If Ghostscript installation has been requested, the location of `gs5.50` is added to your `PATH` because the files `gswin32c.exe` (command line interface to Ghostscript) and `gsdll32.dll` (dll embedded Ghostscript) are accessed by several programs of the distribution. Ghostscript uses the registry now (version 5.50) and so does not need any other setting to find the relevant files. Previous or customized installations might require to set the `GS_LIB` variable. See the appropriate documentation.

ImageMagick is also added to your `PATH` if it has been selected. Moreover, the right *delegates.mk* file is copied according to your platform (Windows 9x or Windows NT). See *ImageMagick* documentation for more details.

Your main `<root>/texmf/web2c/texmf.cnf` file is edited to reflect the `texmf` trees you have specified, and the location of locally generated fonts. The variables `$TEXMF`, `$TEXMFLOCAL`, `$HOMETEXMF`, and `$VARTEXFONTS` are modified.

The file `<root>/texmf/web2c/mktex.cnf` is edited to add the feature 'varfonts', which will force any locally generated font to be stored in `$VARTEXFONTS`.

The configuration for tex4ht is undertaken by editing `<root>/texmf/tex4ht/base`. The only relevant part is that it needs to run the `convert.exe` tool of *ImageMagick*, so the full path to access it is provided.

4.6 Testing the installation

A valuable tool to test the installation now is the program `kpsewhich`.

As a first step, you should check if Web2C correctly identifies the location of your `texmf` tree. Open a command prompt window and type

```
kpsewhich -expand-path=$TEXMF
```

The answer should be the location of your `texmf` tree (e.g., `c:/TeX/texmf` if you unpacked the archive files as in the example above —note that the answer is a Unix style path, i.e., the MS-Dos style `\\` is substituted by `/`; you don't have to worry about this). If you do not get the right answer you have probably changed the default directory structure. In this case you have to set the variable `TEXMF` manually to the root directory of your `texmf` tree.

If you want to be on the safe side, you may type in `mktexlsr` to update the `ls-R` database, even though a proper `ls-R` file should be provided after installation.

4.7 Network installation and filesystem considerations

All the support files, everything except the files in `bin/win32`, are shareable with a Unix installation. This means you can use Samba either to mount from a Windows NT server to a Unix workstation or the converse. Several strategies are possible:

- Put everything on the server. Just add each set of files for the operating system and architecture you want to use in the `bin` directory. That means for example `bin/win32` and `bin/i386-linux`.
- Install a local copy of the binaries and format files. In this case, assign `$TEXMFMAIN` to the main `texmf` tree that will lie on the network.

These schemes should have been handled by the InstallShield installer. But so many problems arose with this installer that these features have been delayed to the next version of the setup program.

Win32 supports multiple filesystems:

- MS-Dos FAT, 8.3 and uppercase filenames
- Protected mode FAT, long filenames, but case-insensitive
- NTFS, long filenames and case-sensitive
- ISO9660 CD-ROM, 8.3 and uppercase filenames

Moreover, Win32 calls which refer to filenames are case-insensitive, and there are several other features in NTFS that Win32 can't use for the moment. Another dimension is the use of different directory separators: `/` or `\`, but Win32 calls accept both.

So what difficulties may arise ?

Most likely, you will have some style files with long filenames. If you are running on a filesystem which supports them,[5] there is no problem and you have nothing to do. Otherwise, you will need to use the alias feature of Kpathsea. Suppose, for instance, you are trying to install `texmf` on a FAT partition and you have the style file named `longtable.sty` in your tree. The filename will be truncated to its 8.3 form: `longtabl.sty`. In this case, you will need to create a file named `aliases` in the same place as the `ls-R` file in your `texmf` tree. This file should contain the following line:

```
longtabl.sty longtable.sty
```

All references to `longtable.sty` will be redirected to `longtabl.sty` if the long filename is not found.

Otherwise, if you think you have trouble with filenames, consider doing the following:

- paths in config files and environment variables should be written using `/` rather than `\`;
- `ls-R` databases should be in lower case, even if you are running on FAT or CD-ROM;
- use the debug feature of Kpathsea and `kpsewhich` to demonstrate your problem and email us the results of your investigations.

5 Building on a new Unix 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 `source` on the CD-ROM.

You should first install the support tree from the TEX Live CD-ROM (do a basic install, with no system binaries chosen).

[5] For example, NTFS but not FAT!

5.1 Prerequisites

You will need about 100 megabytes of disk space to compile all of TEX and its support programs. You'll also need an ANSI C compiler, a make utility, a lexical scanner, and a parser generator. The GNU utilities (gcc, GNU make, m4, flex, bison) are the most widely tested on different platforms. gcc-2.7.* flex-2.4.7 and GNU make-3.72.1 or newer should work well. 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. The command `uname` must return a sensible value.

5.2 Configuration

First, unpack the source from the compressed `tar` file in the directory `source` to your disk and change directory to where you placed it. Decide where the 'root' of the installation will be, e.g., `/var/TeX` or `/usr/TeX`. Obviously you should use the same location that you specified when you installed the support tree.

Now, start the build process by running configure with a command-line like

```
>> ./configure -prefix=/usr/TeX
```

The 'prefix' directory is the one where you installed the support tree; the directory layout that will be used is as follows (where $TEXDIR stands for the directory you chose):

`$TEXDIR/man`	Unix manual pages
`$TEXDIR/share/texmf`	main tree with fonts, macros, etc
`$TEXDIR/info`	GNU 'info' manuals
`$TEXDIR/bin/$PLATFORM`	binaries

You can omit the use of '`share/`' part for the `texmf` directory if you want, as `$TEXDIR/share/texmf` and `$TEXDIR/texmf` are auto-detected by configure. If you choose something different, you have to specify that directory with the `--datadir` option of configure.

If you want to leave out the `$PLATFORM` directory level (that is, put binaries directly into `$TEXDIR/bin`), you can use the configure option `--disable-multiplatform`.

Have a look at the output of `./configure --help` for more options you can use (such as omitting optional packages such as Ω or ε-TEX).

5.3 Running make

Make sure the shell variable `noclobber` is not set, and then type

```
>> make world
```

and relax. . . .

It could also be useful to log all the output, e.g., by typing

```
>> sh -c "make world >world.log 2>&1" &
```

Before you think that everything is okay, please check the log file for errors (GNU make always uses the string "`Error:`" whenever a command returns an error code) and check if all binaries are built:

```
>> cd /usr/TeX/bin/i586-pc-linux-gnu
>> ls | wc
```

The result should be 204. `make world` is equivalent to `make all install strip`

If you need special privileges for `make install`, you can run two make jobs in separate runs:

```
>> make all
>> su
>> make install strip
```

5.4 Final configuration steps

Set up your `PATH` to include the directory containing the just-installed binaries (e.g., `/usr/TeX/bin/mips-sgi-irix6.3`); similarly, `MANPATH` and `INFOPATH` to include the relevant newly installed subdirectories, i.e., `$TEXDIR/man` and `$TEXDIR/info`.

The program texconfig allows you to set the defaults for hyphenation, paper size, print command, METAFONT mode, etc. You can run this command interactively and see what options it offers, or type

```
>> texconfig help
```

For example, if you are not using A4 format paper, you can make 'lettersize' the default using:

```
>> texconfig dvips paper letter
>> texconfig xdvi paper us
```

6 A user's guide to the Web2c system

Web2c contains a set of TEX-related programs, i.e., TEX itself, METAFONT, MetaPost, BIBTEX, etc; it works on Unix, Windows 3.1, 9x/NT, DOS, 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 core TEX family components are:

bibtex Maintaining bibliographies;

dmp troff to MPX (MetaPost pictures);

dvicopy Produces modified copy of DVI file;

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 7.3 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.

6.1 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 page 31) 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 sought.

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 it transforms 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.

6.1.1 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

```
TEXMF              = {$TEXMFLOCAL;!!$TEXMFMAIN}
TEXINPUTS.latex    = .;$TEXMF/tex/{latex;generic;}//
TEXINPUTS.fontinst = .;$TEXMF/tex//;$TEXMF/fonts/afm//
% e-TeX related files
TEXINPUTS.elatex   = .;$TEXMF/{etex;tex}/{latex;generic;}//
TEXINPUTS.etex     = .;$TEXMF/{etex;tex}/{eplain;plain;generic;}//
```

Figure 4: An illustrative configuration file sample

being run, then `TEXINPUTS.latex` will override `TEXINPUTS`.

2. A program-specific configuration file, for example, a line "`S /a:/b`" in dvips's `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 page 33).

6.1.2 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 `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 appendix 7, starting on page 37, 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 has the form:

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

  where the "`=`" and surrounding whitespace are 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. A "`;`" in "*`value`*" is translated to "`:`" if running under Unix; this is useful to be able to have a single `texmf.cnf` for Unix, MS-Dos and Windows systems.
- All definitions are read before anything is expanded, so variables can be referenced before they are defined.

A configuration file fragment illustrating most of these points is shown in Figure 4.

6.1.3 Path expansion

Kpathsea recognizes certain special characters and constructions in search paths, similar to those available in Unix shells. As a general example, the 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.

6.1.4 Default expansion

If the highest-priority search path (see "Path sources" on page 29) 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 "`:`".

6.1.5 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 (on one line!):

```
TEXMF = {$HOMETEXMF,$TEXMFLOCAL,
            !!$VARTEXMF,!!$TEXMFMAIN}
```

Using this you can then write something like

```
TEXINPUTS = .;$TEXMF/tex//
```

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

6.1.6 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.

6.1.7 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 (can occur anywhere in a path, except at its begining).

`%` Start of comment.

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

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

6.2 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 DOS-like "8.3" filename conventions in source files.

6.2.1 The 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 hierarchy in your installation that you wish to be searched (`$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 `mktexlsr` script included with the distribution. It is invoked by the various "`mktex`"... 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.

6.2.2 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 "`mktex`"... scripts).

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

The options specified in "*`option`*" 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. For formats which do not have an associated unambiguous suffix, such as MetaPost support files and dvips configuration files, you have to specify the name as found in the first column of Table 1 on p. 36, which lists currently recognized names, a description, associated environment variables,[6], and possible file extensions.

The last two entries in Table 1 are special cases, where the paths and environment variables depend on the name of the program: the variable name is constructed by converting the program name to upper case, and then appending `INPUTS`.

The 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.

`--mode=`*`string`*
Set the mode name to "*`string`*"; this 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 "*`.prognam`*" 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 a name can be used, just as with "`--format`" option.

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

6.2.3 Examples of use

Let us now have a look at Kpathsea in action.

```
>> kpsewhich  article.cls
/usr/texmf/tex/latex/base/article.cls
```

We are looking for the file `article.cls`. Since the "`.cls`" suffix is unambiguous we do not need to specify that we want to look for a file of type "tex" (TeX source file directories). We find it in the subdirectory `tex/latex/base` below the "`TEXMF`" root directory. Similarly, all of the following are found without problems thanks to their unambiguous suffix.

```
>> kpsewhich array.sty
   /usr/texmf/tex/latex/tools/array.sty
>> kpsewhich latin1.def
   /usr/texmf/tex/latex/base/latin1.def
>> kpsewhich size10.clo
   /usr/texmf/tex/latex/base/size10.clo
>> kpsewhich small2e.tex
   /usr/texmf/tex/latex/base/small2e.tex
>> kpsewhich tugboat.bib
   /usr/texmf/bibtex/bib/beebe/tugboat.bib
```

The last item is a BibTeX bibliography database for *TUGBoat* articles.

```
>> kpsewhich cmr10.pk
```

Font bitmap glyph files of type `.pk` are used by display programs like dvips and xdvi. Nothing is returned in this case since there are no pre-generated Computer Modern "`.pk`" files on our system (since we use the Type1 versions on the CD-ROM).

```
>> kpsewhich ecrm1000.pk
/usr/texmf/fonts/pk/ljfour/jknappen/
...                    ec/ecrm1000.600pk
```

For the extended Computer Modern files we had to generate "`.pk`" files, and since the default METAFONT mode on our installation is `ljfour` with a

[6] You can find definitions for these environment variables in the file `texmf.cnf` (page 37)

base resolution of 600 dpi (dots per inch), this instantiation is returned.

```
>> kpsewhich -dpi=300 ecrm1000.pk
```

In this case, when specifying that we are interested in a resolution of 300dpi (-dpi=300) we see that no such font is 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 mktexpk.

Next we turn our attention to dvips's header and configuration files. We first look at one of the commonly used files, 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. As the ".ps" suffix is ambiguous we have to specify explicitly which type we are considering ("dvips config") for the file config.ps.

```
>> kpsewhich tex.pro
   /usr/texmf/dvips/base/tex.pro
>> kpsewhich --format="dvips config" config.ps
   /usr/texmf/config/config.ps
>> kpsewhich psfonts.map
   /usr/texmf/dvips/base/psfonts.map
```

We now take a closer look 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="dvips config" config.utm
/usr/texmf/dvips/psnfss/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 --format="dvips config" utm.map
   /usr/texmf/dvips/psnfss/utm.map
```

This map file defines the file names of the Type1 PostScript fonts in the URW collection. Its contents look like (we only show some 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 us, for instance, take the Times Regular instance utmr8a.pfb and find its position in the texmf directory tree by using a search for Type1 font files:

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

It should be evident from these few examples how you 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.

6.2.4 Debugging actions

Sometimes it is necessary to 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, by setting a combination of debug switches, one can follow in detail where files are being 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, one 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 uses only the font cmr10, so let us look how dvips prepares the PostScript file (we want to use the Type1 version of the Computer Modern fonts, hence the option -Pcms).

```
>> dvips -d4100 hello-world -Pcms -o
```

In this case we have combined dvips's debug class 4 (font paths) with Kpathsea's path element expansion (see dvips Reference Manual, texmf/doc/html/dvips/dvips_toc.html). The output (slightly rearranged) appears in Figure 5. dvips starts by locating its working files. First, texmf.cnf is found, which gives the definitions of the search paths for the other files, then the file database ls-R (to optimize file searching) and the file aliases, which makes it possible to declare several names (e.g., a short DOS-like "8.3" and a more natural longer version) for the same file. Then dvips goes on to find the generic configuration file config.ps before looking for the customization file .dvipsrc (which, in this case is *not found*). Finally, dvips locates the config file for the Computer Modern PostScript fonts config.cms (this was initiated with the -Pcms option on the dvips command). This file contains the list of the "map" files which define the relation between the TeX, PostScript and file system names of the fonts.

```
>> more /usr/texmf/dvips/cms/config.cms
   p +ams.map
   p +cms.map
   p +cmbkm.map
   p +amsbkm.map
```

dvips thus goes on to find all these files, plus the generic map file psfonts.map, which is always loaded (it contains declarations for commonly used PostScript fonts; see the last part of Section 6.2.3 for more details about PostScript map file handling).

At this point dvips identifies itself to the user...

```
This is dvips 5.78 Copyright 1998 Radical Eye...
```

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

After having found the file in question, 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 1998.02.26:1204’ -> hello-world.ps
Defining font () cmr10 at 10.0pt
Font cmr10 <CMR10> is resident.
```

Now the search (Figure 8) starts for cmr10.tfm, which is found, then a few more prolog files (not shown) are referenced, and finally the Type1 instance cmr10.pfb of the font is located and included in the output file (see last line).

6.3 Runtime options

Another of the nice features of Web2c 7.3 is its possibility to control a number of memory parameters (in particular, array sizes) via the runtime file texmf.cnf read by Kpathsea. The listing of texmf.cnf is shown in Section 7, starting on page 37; the settings of all parameters can be found in Part 3 of that file. The more important control variables 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 (compare the generic value and the "huge" one instantiated by hugetex, etc.).

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. You can see that both the hugetex and pdflatex program invocations ask for an extra 15,000 control sequences (the default value of hash_extra is zero).

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.

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

Figure 5: Finding configuration files

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

Figure 6: Finding the prolog file

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

Figure 7: Finding the font file

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

Figure 8: Finding the Type 1 font file

Table 1: Kpathsea file types

Name	*Description*	*Variables*	*Suffixes*
afm	Adobe font metrics	AFMFONTS	.afm
base	Metafont memory dump	MFBASES, TEXMFINI	.base
bib	BibTeX bibliography source	BIBINPUTS, TEXBIB	.bib
	bitmap fonts	GLYPHFONTS, TEXFONTS	
bst	BibTeX style files	BSTINPUTS	.bst
cnf	Runtime configuration files	TEXMFCNF	.cnf
dvips config	dvips configuration files, e.g., config.ps and psfonts.map	TEXCONFIG	.map
fmt	TeX memory dump	TEXFORMATS, TEXMFINI	.fmt, .efmt, .efm
gf	generic font bitmap	GFFONTS	.gf
graphic/figure	Encapsulated PostScript figures	TEXPICTS, TEXINPUTS	.eps, .epsi
ist	makeindex style files	TEXINDEXSTYLE, INDEXSTYLE	.ist
ls-R	Filename databases	TEXMFDBS	
map	Fontmaps	TEXFONTMAPS	.map
mem	MetaPost memory dump	MPMEMS, TEXMFINI	.mem
mf	Metafont source	MFINPUTS	.mf
mfpool	Metafont program strings	MFPOOL, TEXMFINI	.pool
mft	MFT style file	MFTINPUTS	.mft
	miscellaneous fonts	MISCFONTS	
mp	MetaPost source	MPINPUTS	.mp
mppool	MetaPost program strings	MPPOOL, TEXMFINI	.pool
MetaPost support	MetaPost support files, used by DMP	MPSUPPORT	
ocp	Ω compiled process files	OCPINPUTS	.ocp
ofm	Ω font metrics	OFMFONTS, TEXFONTS	.ofm, .tfm
opl	Ω property lists	OPLFONTS, TEXFONTS	.opl
otp	Ω translation process files	OTPINPUTS	.otp
ovf	Ω virtual fonts	OVFFONTS, TEXFONTS	.ovf
ovp	Ω virtual property lists	OVPFONTS, TEXFONTS	.ovp
pk	packed bitmap fonts	*program*FONTS (*program* being XDVI, etc.), PKFONTS, TEXPKS, GLYPHFONTS, TEXFONTS	.pk
PostScript header	downloadable PostScript	TEXPSHEADERS, PSHEADERS	.pro, .enc
tex	TeX source	TEXINPUTS	.tex, .cls, .sty, .clo, .def
TeX system documentation	Documentation files for the TeX system	TEXDOCS	
TeX system sources	Source files for the TeX system	TEXSOURCES	
texpool	TeX program strings	TEXPOOL, TEXMFINI	.pool
tfm	TeX font metrics	TFMFONTS, TEXFONTS	.tfm
Troff fonts	Troff fonts, used by DMP	TRFONTS	
truetype fonts	TrueType outline fonts	TTFONTS	.ttf, .ttc
type1 fonts	Type 1 PostScript outline fonts	T1FONTS, T1INPUTS, TEXPSHEADERS, DVIPSHEADERS	.pfa, .pfb
type42 fonts	Type 42 PostScript outline fonts	T42FONTS	
vf	virtual fonts	VFFONTS, TEXFONTS	.vf
web2c files	Web2c support files	WEB2C	
other text files	text files used by ‘foo’	FOOINPUTS	
other binary files	binary files used by ‘foo’	FOOINPUTS	

7 The texmf.cnf file

```
% TeX Live texmf.cnf
%  Part 1: Search paths and directories.
%
% You can set an environment variable to override TEXMF 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.

% The main tree, which must be mentioned in $TEXMF, below:
TEXMFMAIN = $SELFAUTOPARENT/texmf

% A place for local additions to a "standard" texmf tree.
TEXMFLOCAL = $SELFAUTOPARENT/texmf-local

% User texmf trees can be catered for like this...
HOMETEXMF=$HOME/texmf

% A place where texconfig stores modifications (instead of the TEXMFMAIN
% tree).  texconfig relies on the name, so don’t change it.
VARTEXMF = $SELFAUTOPARENT/texmf-var

% Now, list all the texmf trees. If you have multiple trees,
% use shell brace notation, like this:
%   TEXMF = {$HOMETEXMF,!!$VARTEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN}
% The braces are necessary.
TEXMF = {$HOMETEXMF,$TEXMFLOCAL,!!$VARTEXMF,!!$TEXMFMAIN}

% The system trees.  These are the trees that are shared by all the users.
SYSTEXMF = $TEXMF

% Where generated fonts may be written.  This tree is used when the sources
% were found in a system tree and either that tree wasn’t writable, or the
% varfonts feature was enabled in MT_FEATURES in mktex.cnf.
VARTEXFONTS = /var/tmp/texfonts

% 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 = $TEXMF;$VARTEXFONTS

% It may be convenient to define TEXMF like this:
%   TEXMF = {$HOMETEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN,$HOME}
% which allows users to set up entire texmf trees, and tells TeX to
% look in places like ~/tex and ~/bibtex.  If you do this, define TEXMFDBS
% like this:
%   TEXMFDBS = $HOMETEXMF;$TEXMFLOCAL;$TEXMFMAIN;$VARTEXFONTS
% or mktexlsr will generate an ls-R file for $HOME when called, which is
% rarely desirable.  If you do this you’ll want to define SYSTEXMF like
% this:
%   SYSTEXMF = $TEXMFLOCAL;$TEXMFMAIN
% so that fonts from a user’s tree won’t escape into the global trees.
%
% On some systems, there will be a system tree which contains all the font
% files that may be created as well as the formats.  For example
%   VARTEXMF = /var/lib/texmf
% is used on many Linux systems.  In this case, set VARTEXFONTS like this
%   VARTEXFONTS = $VARTEXMF/fonts
% and do not mention it in TEXMFDBS (but _do_ mention VARTEXMF).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Usually you will not need to edit any of the other variables in part 1. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% WEB2C is for Web2C specific files.  The current directory may not be
% a good place to look for them.
WEB2C = $TEXMF/web2c

% TEXINPUTS is for TeX input files -- i.e., anything to be found by \input
% or \openin, including .sty, .eps, etc.

% LaTeX-specific macros are stored in latex.
TEXINPUTS.latex = .;$TEXMF/tex/{latex,generic,}//
TEXINPUTS.hugelatex = .;$TEXMF/tex/{latex,generic,}//

% Fontinst needs to read afm files.
TEXINPUTS.fontinst = .;$TEXMF/{tex{/fontinst,},fonts/afm}//

% Plain TeX.  Have the command tex check all directories as a last
% resort, we may have plain-compatible stuff anywhere.
TEXINPUTS.tex = .;$TEXMF/tex/{plain,generic,}//
% other plain-based formats
TEXINPUTS.amstex = .;$TEXMF/tex/{amstex,plain,generic,}//
TEXINPUTS.ftex = .;$TEXMF/tex/{formate,plain,generic,}//
TEXINPUTS.texinfo = .;$TEXMF/tex/{texinfo,plain,generic,}//
TEXINPUTS.eplain  = .;$TEXMF/tex/{eplain,plain,generic,}//
TEXINPUTS.jadetex = .;$TEXMF/tex/{jadetex,generic,plain,}//
TEXINPUTS.pdfjadetex = .;$TEXMF/{pdftex,tex}/{jadetex,generic,plain,}//

% e-TeX.
TEXINPUTS.elatex   = .;$TEXMF/{etex,tex}/{latex,generic,}//
TEXINPUTS.etex     = .;$TEXMF/{etex,tex}/{generic,plain,}//

% PDFTeX.  This form of the input paths is borrowed from teTeX.  A certain
% variant of TDS is assumed here, unaffected by the build variables.
TEXINPUTS.pdftexinfo = .;$TEXMF/{pdftex,tex}/{texinfo,plain,generic,}//
TEXINPUTS.pdflatex = .;$TEXMF/{pdftex,tex}/{latex,generic,}//
TEXINPUTS.pdftex   = .;$TEXMF/{pdftex,tex}/{plain,generic,}//
TEXINPUTS.pdfelatex = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{latex,generic,}//
TEXINPUTS.pdfetex   = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{plain,generic,}//

% Omega.
TEXINPUTS.lambda = .;$TEXMF/{omega,tex}/{lambda,latex,generic,}//
TEXINPUTS.omega = .;$TEXMF/{omega,tex}/{plain,generic,}//

% Context macros by Hans Hagen:
TEXINPUTS.context   = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{context,plain,generic,}//

% cstex, from Petr Olsak
TEXINPUTS.cslatex = .;$TEXMF/tex/{cslatex,csplain,latex,generic,}//
TEXINPUTS.csplain = .;$TEXMF/tex/{csplain,plain,generic,}//
TEXINPUTS.pdfcslatex = .:$TEXMF/{pdftex,tex}/{cslatex,csplain,latex,generic,}//
TEXINPUTS.pdfcsplain = .:$TEXMF/{pdftex,tex}/{csplain,plain,generic,}//

% Polish
TEXINPUTS.platex  = .;$TEXMF/tex/{platex,latex,generic,}//
TEXINPUTS.pdfmex   = .;$TEXMF/{pdftex,tex}/{mex,plain,generic,}//
TEXINPUTS.mex   = .;$TEXMF/tex/{mex,plain,generic,}//
```

```
% french
TEXINPUTS.frtex   = .;$TEXMF/{mltex,tex}/{plain,generic,}//
TEXINPUTS.frlatex = .;$TEXMF/{mltex,tex}/{frlatex,latex,generic,}//

% MLTeX
TEXINPUTS.mltex = .;$TEXMF/{mltex,tex}/{plain,generic,}//
TEXINPUTS.mllatex = .;$TEXMF/{mltex,tex}/{latex,generic,}//

% odd formats  needing their own paths
TEXINPUTS.lollipop = .;$TEXMF/tex/{lollipop,generic,plain,}//
TEXINPUTS.lamstex  = .;$TEXMF/tex/{lamstex,generic,plain,}//

% Earlier entries override later ones, so put this last.
TEXINPUTS = .;$TEXMF/tex/{generic,}//

% Metafont, MetaPost inputs.
MFINPUTS = .;$TEXMF/metafont//;{$TEXMF/fonts,$VARTEXFONTS}/source//
MPINPUTS = .;$TEXMF/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 = .;$TEXMF/web2c
MFBASES = .;$TEXMF/web2c
MPMEMS = .;$TEXMF/web2c
TEXPOOL = .;$TEXMF/web2c
MFPOOL = .;$TEXMF/web2c
MPPOOL = .;$TEXMF/web2c

% Device-independent font metric files.
VFFONTS = .;$TEXMF/fonts/vf//
TFMFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/tfm//

% The $MAKETEX_MODE below means the drivers will not use a cx font when
% the mode is ricoh.  If no mode is explicitly specified, kpse_prog_init
% sets MAKETEX_MODE to /, so all subdirectories are searched.  See the manual.
% The modeless part guarantees that bitmaps for PostScript fonts are found.
PKFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/pk/{$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 = .;$TEXMF/fonts/gf/$MAKETEX_MODE//

% A backup for PKFONTS and GFFONTS. Not used for anything.
GLYPHFONTS = .;$TEXMF/fonts

% For texfonts.map and included map files used by mktexpk.
% See ftp://ftp.tug.org/tex/fontname.tar.gz.
TEXFONTMAPS = .;$TEXMF/fontname

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

% PostScript headers, prologues (.pro), encodings (.enc) and fonts;
% this is also where pdftex finds included figures files!
```

```
TEXPSHEADERS.pdflatex   = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdfelatex  = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdftexinfo = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdfcslatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdfcsplain = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdfetex    = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdfjadetex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdfmex     = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdftex     = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.pdftexinfo = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.cont-de    = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.cont-en    = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.cont-nl    = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS.context    = .;$TEXMF/{etex,tex,pdftex,dvips,fonts/type1}//
TEXPSHEADERS = .;$TEXMF/{dvips,fonts/type1,pdftex}//

% PostScript Type 1 outline fonts.
T1FONTS = .;$TEXMF/fonts/type1//

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

% TrueType outline fonts.
TTFONTS = .;$TEXMF/fonts/truetype//

% Type 42 outline fonts.
T42FONTS = .;$TEXMF/fonts/type42//

% A place to puth everything that doesn’t fit the other font categories.
MISCFONTS = .;$TEXMF/fonts/misc//

% Dvips’ config.* files (this name should not start with ‘TEX’!).
TEXCONFIG = .;$TEXMF/dvips//

% Makeindex style (.ist) files.
INDEXSTYLE = .;$TEXMF/makeindex//

% Used by DMP (ditroff-to-mpx), called by makempx -troff.
TRFONTS = /usr/lib/font/devpost
MPSUPPORT = .;$TEXMF/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 = c:/TeX/etc
MAILCAPLIBDIR = c:/TeX/etc

% TeX documentation and source files, for use with kpsewhich.
TEXDOCS = .;$TEXMF/doc//
TEXSOURCES = .;$TEXMF/source//

% Omega-related fonts and other files.  The odd construction for OFMFONTS
% makes it behave in the face of a definition of TFMFONTS.  Unfortunately
% no default substitution would take place for TFMFONTS, so an explicit
% path is retained.
OFMFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/{ofm,tfm}//;$TFMFONTS
OPLFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/opl//
OVFFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/ovf//
OVPFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/ovp//
```

```
OTPINPUTS = .;$TEXMF/omega/otp//
OCPINPUTS = .;$TEXMF/omega/ocp//

%% TeX4ht utility, sharing files with TeX4ht
T4HTINPUTS = .;$TEXMF/tex4ht//

%% The mktex* scripts rely on KPSE_DOT. Do not set it in the environment.
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 /$VARIABLE, we’d end up with // if VARIABLE is defined,
%    which would search the entire world.
%
% The TETEXDIR stuff isn’t likely to be relevant unless you’re using teTeX,
% but it doesn’t hurt.
%
TEXMFCNF = .;{$SELFAUTOLOC,$SELFAUTODIR,$SELFAUTOPARENT}\
{,{/share,}/texmf{.local,}/web2c};c:/TeX/texmf/web2c




%  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 = none

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

% Allow TeX \openout/\openin on filenames starting with ‘.’ (e.g., .rhosts)?
% a (any)        : any file can be opened.
% r (restricted) : disallow opening "dotfiles".
% p (paranoid)   : as ’r’ and disallow going to parent directories, and
%                  restrict absolute paths to be under $TEXMFOUTPUT.
openout_any = p
openin_any = a
% Allow TeX, MF, and MP to parse the first line of an input file for
% the %&format construct.
parse_first_line = t

% Enable the mktex... 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 ‘MKTEXPK’ value.
%
% MKTEXTEX = 0
% MKTEXPK = 0
% MKTEXMF = 0
% MKTEXTFM = 0
% MKOCP = 0
% MKOFM = 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 total.
%
% 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 extra_mem_*.  (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.
%
% For some xy-pic samples, you may need as much as 700000 words of memory.
% For the vast majority of documents, 60000 or less will do.
%
main_memory = 263000 % words of inimemory available; also applies to inimf&mp
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 = 200000

% Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
font_max = 1000

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

% Max number of characters in all strings, including all error messages,
% help texts, font names, file names, control sequences.
% These values apply to TeX and MP.
pool_size = 125000
```

```

% 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
max_strings = 15000                % 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

% Buffer size.  TeX uses the buffer to contain input lines, but macro
% expansion works by writing material into the buffer and reparsing the
% line.  As a consequence, certain constructs require the buffer to be
% very large.  As distributed, the size is 50000; most documents can be
% handled within a tenth of this size.
buf_size = 50000

% These are Omega-specific.
ocp_buf_size = 20000        % character buffers for ocp filters.
ocp_stack_size = 10000        % stacks for ocp computations.
ocp_list_size = 1000        % control for multiple ocps.

% 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;
% 60 <= max_print_line;
% These apply to Metafont and MetaPost as well.
error_line = 79
half_error_line = 50
max_print_line = 79
stack_size = 300        % simultaneous input sources
save_size = 4000        % for saving values outside current group
param_size = 500        % simultaneous macro parameters
max_in_open = 15        % simultaneous input files and error insertions
hyph_size = 1000        % number of hyphenation exceptions, >610 and <32767
nest_size = 100                % simultaneous semantic levels (e.g., groups)

main_memory.context = 1100000
hash_extra.context = 25000
pool_size.context = 750000
string_vacancies.context = 45000
max_strings.context = 55000
pool_free.context = 47500
nest_size.context = 500
```

```
param_size.context = 1500
save_size.context = 5000
stack_size.context = 1500

main_memory.hugetex = 1100000
param_size.hugetex = 1500
stack_size.hugetex = 1500
hash_extra.hugetex = 15000
string_vacancies.hugetex = 45000
pool_free.hugetex = 47500
nest_size.hugetex = 500
save_size.hugetex = 5000
pool_size.hugetex = 500000
max_strings.hugetex = 55000

main_memory.hugelatex = 1100000
param_size.hugelatex = 1500
stack_size.hugelatex = 1500
hash_extra.hugelatex = 15000
string_vacancies.hugelatex = 45000
pool_free.hugelatex = 47500
nest_size.hugelatex = 500
save_size.hugelatex = 5000
pool_size.hugelatex = 500000
max_strings.hugelatex = 55000

main_memory.jadetex = 1500000
param_size.jadetex = 1500
stack_size.jadetex = 1500
hash_extra.jadetex = 50000
string_vacancies.jadetex = 45000
pool_free.jadetex = 47500
nest_size.jadetex = 500
save_size.jadetex = 5000
pool_size.jadetex = 500000
max_strings.jadetex = 55000

main_memory.pdfjadetex = 2500000
param_size.pdfjadetex = 1500
stack_size.pdfjadetex = 1500
hash_extra.pdfjadetex = 50000
string_vacancies.pdfjadetex = 45000
pool_free.pdfjadetex = 47500
nest_size.pdfjadetex = 500
save_size.pdfjadetex = 5000
pool_size.pdfjadetex = 500000
max_strings.pdfjadetex = 55000

main_memory.pdflatex = 1500000
param_size.pdflatex = 1500
stack_size.pdflatex = 1500
hash_extra.pdflatex = 15000
string_vacancies.pdflatex = 45000
pool_free.pdflatex = 47500
nest_size.pdflatex = 500
pool_size.pdflatex = 500000
save_size.pdflatex = 5000
max_strings.pdflatex = 55000
```

Software

TEX and the Year 2000

Barbara Beeton

During the past year or so, perhaps the most high-profile question in computer circles has been, Will this system continue to work as usual after the clocks strike midnight on 31 December 1999? Although the millennium doesn't really arrive until 2001, the year 2000 (Y2K) is just as important to anyone who depends on a computer. Many so-called "legacy systems" assumed that two digits would be sufficient to accurately represent a year, and standard mainframe tape labels (to cite but one example) designated 99366 as the date (in Julian form) on which data on the tapes could be considered "scratch". To be fair to the designers and programmers of those systems, when this convention was devised, computer memory and storage were very expensive, and the saving in space permitted by this shorthand made it possible to create programs to do things that would otherwise have been prohibitively expensive.

However, the chronometer is about to cycle, and, for two-digit-based systems, how is the year "00" to be interpreted? As 2000 or as 1900?

Some specialties have probably already had to take "real" years into consideration — books were published long before 1900, so libraries might well have holdings from the 1800s or even earlier, if only in special collections. But the bulk of commerce, and governments, are much less farsighted, and system managers and CIOs are carefully auditing their programs and procedures for residual code and assumptions that could raise havoc when the year 2000 arrives.

Is TEX one of the programs that could have Y2K problems? The short answer is, "No, the TEX *program* will not have Y2K problems." However, since this question will be asked at many user sites, it would be helpful to have some guidelines for answering it.

Year handling in TEX

Let's start by examining the code in `tex.web`. This is the canonical source, from which all implementations are ultimately derived. The extracts on the next two pages, where "year" or "date" occurs in either code or commentary, are from version 3.14159, dated 1998/08/08, at `CTAN:systems/knuth/tex/tex.web`.

From these extracts and other code not shown, it can be determined that the primitive `\year` is defined as a 4-byte signed integer. An integer of this size is more than adequate to hold a 4-digit date, or any other value likely to be passed to it from the system or input by the user. It also turns out that the modulo function[1] applied in line 24125 has no effect on any internal value; this is used only to write out information about the preloaded format in the log file. For example, a teTEX implementation on a local Unix machine reports the following:

```
This is TeX, Version 3.14159 (C version 6.1)
 (format=tex 97.10.7)
```

Although one can safely conclude that TEX itself is not subject to the kind of problems that cause Y2K worries, there are two caveats.

- The format message might raise a red flag to someone who is concerned more with form than with substance.
- An application *can* set `\year` to an unconventional value and use it for some operation that might cause a problem.

The second case is a site-dependent problem, not one of TEX; it is up to whoever is performing the local audit to check applications for such use.

The first case, while harmless, is nonetheless amenable to being fixed.

Expanding the TEX format date to report a 4-digit year

The possible ramifications of leaving the format year with two digits was discussed at length on the `tex-implementors` list. The consensus was that, although it is harmless, it would cause questions to be asked and occupy time that would be better spent in tending to real problems. Therefore, after verifying the facts with Peter Breitenlohner, I took advantage of my status as Knuth's "TEX entomologist" and sent him a message asking him to consider another change to TEX and METAFONT (which treats the year in a similar manner).

Within a few days, I received the following answer.

— — ∗ — —

Date: Tue, 24 Nov 1998 00:02:14 -0800 (PST)
Subject: note from Don Knuth

Dear Barbara,

Thanks for your note. It spurred me to [. . .] update my macro files for errata in TAOCP; now they are Y2K compliant!

[1] The modulo function yields the remainder after dividing the operand by a specified value; in the present example, 1999 modulo 100 becomes simply 99.

Extracts from `tex.web` 3.14159 containing “year” or “date”

```
2137:    (Actually there are three places where \TeX\ uses |div| with a possibly
         negative numerator. These are harmless; see |div| in the index. Also if the
         user sets the \.{\\time} or the \.{\\year} to a negative value, some diagnostic
         information will involve negative-numerator division. The same remarks
         apply for |mod| as well as for |div|.)

4918:    @d time_code=20 {current time of day}
         @d day_code=21 {current day of the month}
         @d month_code=22 {current month of the year}
         @d year_code=23 {current year of our Lord}

4981:    @d time==int_par(time_code)
         @d day==int_par(day_code)
         @d month==int_par(month_code)
         @d year==int_par(year_code)

5045:    time_code:print_esc("time");
         day_code:print_esc("day");
         month_code:print_esc("month");
         year_code:print_esc("year");

5130:    primitive("time",assign_int,int_base+time_code);@/
         @!@:time_}{\.{\\time} primitive@>
         primitive("day",assign_int,int_base+day_code);@/
         @!@:day_}{\.{\\day} primitive@>
         primitive("month",assign_int,int_base+month_code);@/
         @!@:month_}{\.{\\month} primitive@>
         primitive("year",assign_int,int_base+year_code);@/
         @!@:year_}{\.{\\year} primitive@>

5218:    @ The following procedure, which is called just before \TeX\ initializes its
         input and output, establishes the initial values of the date and time.
         @^system dependencies@>
         Since standard \PASCAL\ cannot provide such information, something special
         is needed. The program here simply specifies July 4, 1776, at noon; but
         users probably want a better approximation to the truth.

         @p procedure fix_date_and_time;
         begin time:=12*60; {minutes since midnight}
         day:=4; {fourth day of the month}
         month:=7; {seventh month of the year}
         year:=1776; {Anno Domini}
         end;

10292:   @<Print the banner line, including the date and time@>;

10322:   @ @<Print the banner...@>=
         begin wlog(banner);
         slow_print(format_ident); print("  ");
         print_int(day); print_char(" ");
         months:='JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC';
         for k:=3*month-2 to 3*month do wlog(months[k]);
         print_char(" "); print_int(year); print_char(" ");
         print_two(time div 60); print_char(":"); print_two(time mod 60);
         end
```

```
11630:  @ The preamble contains basic information about the file as a whole. As
        stated above, there are six parameters:
        $$\hbox{|@!i[1]| |@!num[4]| |@!den[4]| |@!mag[4]| |@!k[1]| |@!x[k]|.}$$
        The |i| byte identifies \.{DVI} format; currently this byte is always set
        to~2. (The value |i=3| is currently used for an extended format that
        allows a mixture of right-to-left and left-to-right typesetting.
        Some day we will set |i=4|, when \.{DVI} format makes another
        incompatible change---perhaps in the year 2048.)
```

```
12252:  @ @<Initialize variables as |ship_out| begins@>=
        dvi_h:=0; dvi_v:=0; cur_h:=h_offset; dvi_f:=null_font;
        ensure_dvi_open;
        if total_pages=0 then
          begin dvi_out(pre); dvi_out(id_byte); {output the preamble}
        @^preamble of \.{DVI} file@>
          dvi_four(25400000); dvi_four(473628672); {conversion ratio for sp}
          prepare_mag; dvi_four(mag); {magnification factor is frozen}
          old_setting:=selector; selector:=new_string;
          print(" TeX output "); print_int(year); print_char(".");
          print_two(month); print_char("."); print_two(day);
          print_char(":"); print_two(time div 60);
          print_two(time mod 60);
          selector:=old_setting; dvi_out(cur_length);
          for s:=str_start[str_ptr] to pool_ptr-1 do dvi_out(so(str_pool[s]));
          pool_ptr:=str_start[str_ptr]; {flush the current string}
          end
```

```
23668:  The global variable |format_ident| is a string that is printed right
        after the |banner| line when \TeX\ is ready to start. For \.{INITEX} this
        string says simply ‘\.{(INITEX)}’; for other versions of \TeX\ it says,
        for example, ‘\.{(preloaded format=plain 82.11.19)}’, showing the year,
        month, and day that the format file was created. We have |format_ident=0|
        before \TeX’s tables are loaded.
```

```
24122:  @ @<Create the |format_ident|...@>=
        selector:=new_string;
        print(" (preloaded format="); print(job_name); print_char(" ");
        print_int(year mod 100); print_char(".");
        print_int(month); print_char("."); print_int(day); print_char(")");
```

```
24214:  initialize; {set global variables to their starting values}
        @!init if not get_strings_started then goto final_end;
        init_prim; {call |primitive| for each primitive}
        init_str_ptr:=str_ptr; init_pool_ptr:=pool_ptr; fix_date_and_time;
        tini@/
```

```
24358:  @<Get the first line...@>=
        begin @<Initialize the input routines@>;
        if (format_ident=0)or(buffer[loc]="&") then
          begin if format_ident<>0 then initialize; {erase preloaded format}
          if not open_fmt_file then goto final_end;
          if not load_fmt_file then
            begin w_close(fmt_file); goto final_end;
            end;
          w_close(fmt_file);
          while (loc<limit)and(buffer[loc]=" ") do incr(loc);
          end;
        if end_line_char_inactive then decr(limit)
        else  buffer[limit]:=end_line_char;
        fix_date_and_time;@/
```

Peter Breitenlohner's comments are 100% correct. Also, I might note that the METAFONT documentation already mentions that METAFONT (as it stands) cannot be run after the year 32767; the latter "problem" is not applicable however to TeX.

I agree that it would now be best to remove the "mod 100" from TeX module 1328 and from MF module 1200 (and from METAPOST in the corresponding place). I hereby give permission to implementors to make such changes in their change files. No change to the version numbers are needed.

I've actually made the changes in my personal copies of `tex.web` and `mf.web` and `tex82.bug` and `mf85.bug` and `errata.eight` and `errata.tex`; but those sources won't be updated at labrea until 2002.

I'm not changing page 23 of *The TeXbook* — where the example was probably my original motivation for cutting to two digits, since the example wouldn't fit on a single line otherwise — nor the corresponding page of *The METAFONTbook*. The format-identifier details are not an essential part of TeX's actions.

Incidentally, I'm not considering this to be the "final bug" in TeX. But it may well turn out to be the final change ever made.

Cordially, Don

— — * — —

Thus all implementors now have permission to make this change — it is official.

Status of TeX implementations

While most implementors have probably made this change by now, I can confirm the changes for only those who have announced it publicly.

Most commercial implementors have done so, and have posted confirmation on their Web sites.

— — * — —

TrueTeX. At `http://truetex.com/y2k.htm`, there is a general statement of the relevant issues.

> TeX Year 2000 (Y2K) Issues in Summary
>
> This page summarizes the year 2000 issues surrounding TeX and METAFONT, based on discussions in the newsgroup `comp.text.tex` and among the members of the mailing list `tex-implementors@ams.org`. We attempt herein to merely set forth the chief matters at hand, without engaging the controversial aspects of what solutions should be taken.
>
> 1. Crashing: The programs TeX and METAFONT themselves will not crash due to dates. (However, each executable implementation depends on a run-time library and an operating system, which should be evaluated in this regard.)
> 2. Timestamps: A 2-digit year is (a) printed in logfiles, and (b) stored in format file and base file time stamps. These items should not be of general concern, because they are intended for human readers and not as input to other programs.
>
>   On November 24, 1998, Donald Knuth granted an unusual permission to modify TeX and METAFONT to use 4-digit timestamps (nearly all implementations, such as web2c, had already been doing so), saying:
>
>   > I agree that it would now be best to remove the "mod 100" from TeX module 1328 and from MF module 1200 (and from METAPOST in the corresponding place). I hereby give permission to implementors to make such changes in their change files. No change to the version numbers are needed. *[As reported by Barbara Beeton on the `tex-implementors` e-mail list.]*
>
>   This permission means that 4-digit timestamps, while changing the output of TeX and METAFONT slightly from the current autographs, still meet Knuth's authoritative standards required of software calling itself TeX or METAFONT.
> 3. The `\year` primitive: TeX TRIP certification, in the strictest sense, does not require that `\year` return a meaningful value (TeX may be certifiably implemented on platforms that do not even supply date reporting, such as standard Pascal). *The TeXbook* does define `\year` as "the current year of our Lord", which is the only correct meaning of `\year` for those implementations which can supply a meaningful value, which is to say nearly all of them.
>
>   In short, TeX implementations should provide a value in `\year` giving a 4-digit year Anno Domini, or the value 1776 if the platform does not support a date function. TeX does not provide any state variables to indicate whether `\year` contains a meaningful value, and while 1776 could have been considered a signal value for a lack of meaning to `\year`, this is not a standardized requirement.
> 4. External software: The TeX corpus embodies many accessory programs, such as macro packages and DVI translators, which may compute dates from the value of `\year` (or rarely, from timestamps). Such accessories should be checked individually for correct behavior when `\year` is assumed to return a correct 4-digit `\year` value before and after

> 2000. Accessories with an additional "defensive level" of correctness will behave reasonably when `\year` contains a two-digit value or a meaningless value such as 1776.

See the TrueTEX home page at `http://truetex.com`.

— — ∗ — —

Other commercial TEX Web sites are more product-specific.

Blue Sky Research. A statement about Macintosh and Textures appears at `http://www.bluesky.com/y2knote.html`:

> **Year 2000 Compliance**
>
> As you may or may not know, the Macintosh in general has been Y2K compliant since it was first introduced in 1984. Contrast this with Microsoft's Windows software — even the Windows 95 version is not Y2K compliant, nor is their Windows NT 4.0! See Apple's year 2000 page on the web.
>
> Likewise, most Macintosh software is built to handle the year 2000, and Textures is no exception. Don Knuth, the author of the TEX typesetting system and the multi-volume set "The Art of Computer Programming" was well ahead of the curve, and TEX has been Y2K compilant [sic.] since the 70's!
>
> In short: Yes, Textures is Y2K compliant.
>
> Apple Macintosh: the choice for the year 2000 and beyond ;-)

— — ∗ — —

PCTEX. At `http://www.pctex.com/iwhatnew.htm`, there is a posting that says

> Is PCTEX Year 2000 Compatible
>
> Because PCTEX and accompanying macros etc. are operating system/date dependant, as long as you have taken proper precautions to ensure your operating system is year 2000 compliant, you will experience no change in operation, come January 1, 2000. If you need a verification letter for your files, click HERE.

The link connects to `ftp://ftp.crl.com/users/rw/pti/98765/yr2000.dvi`.

— — ∗ — —

Y&Y. This statement appears at `http://www.yandy.com/y2k.htm`:

> **Year 2000 (Y2K)**
>
> **Relative to the Year 2000,**
> **there are no date issues with Y&Y TEX System release 2.1**
>
> Y&Y TEX uses only standard calls to the Microsoft C++ library and WIN32 API to manage the dates and times for files. The operating system sets the date and time arithmetic values based on what the PC's realtime clock provides. It is the operating system, the underlying BIOS, and the realtime clock upon which Y&Y software depends for Year 2000 compliance.
>
> We do not embed or implement any algorithm of our own or any third party's to compute dates in any of our products.
>
> TEX itself does not do any computation based on dates. It simply provides informational access to the dates and times obtained from the underlying operating system. The year is not truncated to two digits when offered to the user from inside TEX source code via the controlsequence `\year`, or in information output on screen or in the log file.
>
> In addition, in release 2.1, the date recorded in the format file uses four digits for the year.
>
> This information is provided "AS IS" without warranty of any kind, either express or implied. Any further question may be addressed to the attention of `support@YandY.com`.

— — ∗ — —

MacKichan Software. From the developers of Scientific Word and Scientific WorkPlace (which incorporates TrueTEX), at `http://www.mackichan.com/products/y2k.htm`:

> Dear Sir or Madam:
>
> In response to your inquiry about the MacKichan Software products and the Year 2000 problem, I can give you the following assurances:
>
> **Scientific Notebook**:
> Dates are not used in computations. We have run Scientific Notebook on a system that has had the date advanced beyond the year 2000 and have not experienced any problems. We do not anticipate any Year 2000 issues with this product.
>
> **Scientific Word** and **WorkPlace**:
> **Version 2.5:** Scientific Word and Scientific WorkPlace 2.5 include as a component TrueTEX from Kinch Computer Company. Unfortunately, the version of TrueTEX included with Scientific Word 2.5 and Scientific WorkPlace 2.5 does not operate if you advance the date beyond the year 2000 (see Note below). Version 2.5 is **not** Year 2000 compliant. We expect to post on our web site new versions of the non-Y2K compliant portions of Version 2.5
> **Version 3.0:** We received a new version of TrueTEX from Kinch Computer Company and it does not exhibit a problem if the date is advanced beyond the year 2000. Dates are not used in computations. We have run Scientific Word 3.0 and Scientific WorkPlace 3.0 on a system that has had the date advanced beyond the year 2000 and have not experienced any problems. We do not anticipate any Year 2000 issues with these products.

Note: For the technically inquisitive, Kinch Computer Company reports that the failure of TrueTEX to run properly on a system with the date advanced beyond the year 2000 was corrected after updating to a newer version of the Microsoft C run-time library.

Sincerely,

Barry MacKichan
President

– – ∗ – –

The bottom line

It appears that anyone using the current version of a commercial implementation of TEX can be confident that no untoward results will come from TEX itself.

An article in the latest issue of *NTG MAPS* (Number 22, Voorjaar 1999, pages 136–141) contains some additional information on freeware and shareware implementations of TEX.

Once again, please remember that it would be wise to observe the caution about possible side effects resulting from arithmetic manipulation of year values in *application code* written for use with TEX. Or, as someone wittily observed, they would not like to have to attest to the Y2K implications of David Carlisle's `xii.tex` as published in the last *TUGboat* (19(4), page 348).

Let us go forward confidently into the new year 00, and thence into the new millennium.

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

− − ∗ − −   − − ∗ − −   − − ∗ − −   − − ∗ − −

Hyphenation Exception Log

Barbara Beeton

This is the periodic update[1] of the list of words that TeX fails to hyphenate properly. The list last appeared in *TUGboat* 16, no. 1, starting on page 12. Everything listed there is repeated here. Owing to the length of the list, it has been subdivided into two parts: English words, and names and non-English words that occur in English texts.

This list is specific to the hyphenation patterns that appear in the original `hyphen.tex`, that is, the patterns for U.S. English. If such information for other patterns becomes available, consideration will be given to listing that too. (See below, "Hyphenation for languages other than English".)

In the list below, the first column gives results from TeX's `\showhyphens{...}`; entries in the second column are suitable for inclusion in a `\hyphenation{...}` list.

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

Thanks to all who have submitted entries to the list. Since some suggestions demonstrated a lack of familiarity with the rules of the hyphenation algorithm, here is a short reminder of the relevant idiosyncrasies. Hyphens will not be inserted before the number of letters specified by `\lefthyphenmin`, nor after the number of letters specified by `\righthyphenmin`. For U.S. English, `\lefthyphenmin=2` and `\righthyphenmin=3`; thus no word shorter than five letters will be hyphenated. (For the details, see *The TeXbook*, page 454. For a digression on other views of hyphenation rules, see below under "English hyphenation".) This particular rule is violated in some of the words listed; however, if a word is hyphenated correctly by TeX except for "missing" hyphens at the beginning or end, it has not been included here.

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

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

Words added since the last appearance of the list are preceded by $^+$.

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

English hyphenation

It has been pointed out to me that the hyphenation rules of British English are based on the etymology of the words being hyphenated as opposed to the "syllabic" principles used in the U.S. Furthermore, in the U.K., it is considered bad style to hyphenate a word after only two letters. In order to make TeX defer hyphenation until after three initial letters, set `\lefthyphenmin=3`.

Of course, British hyphenation patterns should be used as well. A set of patterns for UK English has been created by Dominik Wujastyk and Graham Toal, using Frank Liang's `PATGEN` and based on a file of 114925 British-hyphenated words generously made available to Dominik by Oxford University Press. (This list of words and the hyphenation break points in the words are copyright to the OUP and may not be redistributed.) The file of hyphenation patterns may be freely distributed; it is posted on CTAN in the file

```
tex-archive/language/hyphenation/ukhyph.tex
```

and can be retrieved by anonymous FTP or using a Web browser.

Hyphenation for languages other than English

Patterns now exist for many languages other than English, including languages using accented alphabets. CTAN holds an extensive collection of patterns in `tex-archive/language/hyphenation` and its subdirectories.

The List — English words

`academy(ies)`	acad-e-my(ies)
`addable`	add-a-ble
`ad-di-ble`	add-i-ble
`adrenaline`	adren-a-line
`aerospace`	aero-space
`af-terthought`	af-ter-thought
`agronomist`	agron-o-mist

[1] Appeared in *TUGboat* 20(1), March, 1999 on page 50.

	am-phetamine	am-phet-a-mine
	anal-yse(d)	an-a-lyse(d)
	anal-y-ses	analy-ses *
	anomaly(ies)	anom-aly(ies)
	an-tideriva-tive	an-ti-deriv-a-tive
	anti-nomy(ies)	an-tin-o-my(ies)
	antin-u-clear	an-ti-nu-clear
	antin-u-cleon	an-ti-nu-cle-on
	an-tirev-o-lu-tion-ary	an-ti-rev-o-lu-tion-ary
	apotheoses	apoth-e-o-ses
	apotheo-sis	apoth-e-o-sis
	ap-pendix	ap-pen-dix
	archipelago	arch-i-pel-ago
+	archive	ar-chive
+	archiv-ing	ar-chiv-ing
+	archivist	ar-chiv-ist
	archety-pal	ar-che-typ-al
	archetyp-i-cal	ar-che-typ-i-cal
	arc-t-an-gent	arc-tan-gent (better: arc tangent)
	assignable	as-sign-a-ble
	as-sig-nor	as-sign-or
	as-sis-tantship	as-sist-ant-ship
	asymp-tomatic	asymp-to-matic
	asymp-totic	as-ymp-tot-ic
	asyn-chronous	asyn-chro-nous
	atheroscle-ro-sis	ath-er-o-scle-ro-sis
	at-mo-sphere	at-mos-phere
	at-tributed	at-trib-uted
	at-tributable	at-trib-ut-able
+	au-toma-tion	au-to-ma-tion
+	au-toma-ton(a)	au-tom-a-ton(a)
	avoirdupois	av-oir-du-pois
	ban-dleader	band-leader
	bankrupt(cy)	bank-rupt(-cy)
	ba-ronies	bar-onies
	base-li-neskip	\base-line-skip
	bathymetry	ba-thym-e-try
	bathyscaphe	bathy-scaphe
	bea-nies	bean-ies
	be-haviour	be-hav-iour
	be-vies	bevies
	bib-li-og-ra-phystyle	\bib-li-og-ra-phy-style
	bid-if-fer-en-tial	bi-dif-fer-en-tial
	biggest	big-gest
	bil-l-able	bill-able
	biomath-e-mat-ics	bio-math-e-mat-ics
	biomedicine	bio-med-i-cine
	biorhythms	bio-rhythms
	bitmap	bit-map
	blan-der	bland-er
	blan-d-est	bland-est
	blin-der	blind-er
	blon-des	blondes
	blueprint	blue-print
	bornolog-i-cal	bor-no-log-i-cal
	bo-tulism	bot-u-lism
	brus-quer	brusquer
	bus-ier	busier
	bus-i-est	busiest
	buss-ing	bussing
	but-ted	butted
	buz-zword	buzz-word
	ca-cophony(ies)	ca-coph-o-ny(ies)
	cam-er-a-men	cam-era-men
	cartwheel	cart-wheel
	catar-rhs	ca-tarrhs
	catas-trophic	cat-a-stroph-ic
	catas-troph-i-cally	cat-a-stroph-i-cally
	cauliflower	cau-li-flow-er
	cha-parral	chap-ar-ral
	chartreuse	char-treuse
	cholesteric	cho-les-teric
	cigarette	cig-a-rette
	cin-que-foil	cinque-foil
	cognac	co-gnac
	cog-nacs	co-gnacs
+	columns	col-umns
	com-parand	com-par-and
	com-para-nds	com-par-ands
+	com-pendium	com-pen-dium
	comptroller	comp-trol-ler
	con-formable	con-form-able
	con-formist	con-form-ist
	con-for-mity	con-form-ity
	congress	con-gress
	con-tribute(s,d)	con-trib-ute(s,d)
	cose-cant	co-se-cant
	cotan-gent	co-tan-gent
	courses	cour-ses
	crankshaft	crank-shaft
	crocodile	croc-o-dile
	crosshatch(ed)	cross-hatch(ed)
	dachshund	dachs-hund
	database	data-base
	dat-a-p-ath	data-path
+	dat-es-tamp	date-stamp
	declarable	de-clar-able
	defini-tive	de-fin-i-tive
	delectable	de-lec-ta-ble
	democratism	de-moc-ra-tism
	de-mos	demos
	deriva-tive	de-riv-a-tive
	diffract	dif-fract
	di-rer	direr
	di-re-ness	dire-ness
	dis-parand	dis-par-and
	dis-traugh-tly	dis-traught-ly
	dis-tribute(d)	dis-trib-ute(d)
	dou-blespace(ing)	dou-ble-space(-ing)
	dol-lish	doll-ish
	drif-tage	drift-age
+	drivers	driv-ers
	dromedary(ies)	drom-e-dary(ies)
	duopolist	du-op-o-list
	duopoly	du-op-oly
	eas-t-en-ders	east-end-ers
	eco-nomics	eco-nom-ics
	economist	econ-o-mist
	elec-trome-chan-i-cal	electro-mechan-i-cal
	elec-tromechanoa-cous-tic	electro-mechano-acoustic
	eli-tist	elit-ist
	en-trepreneur(ial)	en-tre-pre-neur(-ial)
	epinephrine	ep-i-neph-rine
	equiv-ari-ant	equi-vari-ant
	ethy-lene	eth-yl-ene
	ev-ersible	ever-si-ble
	ev-ert(s,ed,ing)	evert(s,-ed,-ing)

exquisite	ex-quis-ite
ex-traor-di-nary	ex-tra-or-di-nary
fermions	fermi-ons
flag-el-lum(la)	fla-gel-lum(-la)
flammables	flam-ma-bles
fledgling	fledg-ling
flowchart	flow-chart
formidable(y)	for-mi-da-ble(y)
forsythia	for-syth-ia
forthright	forth-right
freeloader	free-loader
friendlier	friend-lier
frivolity	fri-vol-ity
frivolous	friv-o-lous
galac-tic	ga-lac-tic
galaxy(ies)	gal-axy(-ies)
ga-some-ter	gas-om-e-ter
geodesic	ge-o-des-ic
geode-tic	ge-o-det-ic
ge-o-met-ric	geo-met-ric
geotropism	ge-ot-ro-pism
gnomon	gno-mon
grievance	griev-ance
grievous(ly)	griev-ous(-ly)
hairstyle	hair-style
hairstylist	hair-styl-ist
+ half-s-pace	half-space
harbinger	har-bin-ger
harlequin	har-le-quin
hatcheries	hatch-eries
hemoglobin	he-mo-glo-bin
hemophilia	he-mo-phil-ia
hemorhe-ol-ogy	hemo-rhe-ol-ogy
hep-atic	he-pat-ic
hermaphrodite(ic)	her-maph-ro-dite(-ic)
heroes	he-roes
hex-adec-i-mal	hexa-dec-i-mal
holon-omy	ho-lo-no-my
ho-mo-th-etic	ho-mo-thetic
horseradish	horse-rad-ish
hy-potha-la-mus	hy-po-thal-a-mus
ide-als	ideals
ideographs	ideo-graphs
id-iosyn-crasy	idio-syn-crasy
ig-niter	ig-nit-er
ig-n-i-tor	ig-ni-tor
ig-nores-paces	\ignore-spaces
impedances	im-ped-ances
in-finitely	in-fin-ite-ly
in-finites-i-mal	in-fin-i-tes-i-mal
in-fras-truc-ture	in-fra-struc-ture
in-ter-dis-ci-plinary	in-ter-dis-ci-pli-nary
in-ter-galac-tic	in-ter-ga-lac-tic
inu-tile	in-utile
inu-til-ity	in-util-i-ty
ir-re-vo-ca-ble	ir-rev-o-ca-ble
itinerary(ies)	itin-er-ary(-ies)
jeremi-ads	je-re-mi-ads
keystroke	key-stroke
kil-ning	kiln-ing
la-ciest	lac-i-est
lamentable	lam-en-ta-ble
land-sca-per	land-scap-er
larceny(ist)	lar-ce-ny(-ist)
let-terspac-ing	let-ter-spac-ing
lifes-pan	life-span
+ lifestyle	life-style
lightweight	light-weight
limousines	lim-ou-sines
linebacker	line-backer
lines-pac-ing	\line-spacing
+ li-oness	li-on-ess
lithographed	lith-o-graphed
lithographs	lith-o-graphs
lobotomy(ize)	lo-bot-omy(-ize)
lo-ges	loges
longest	long-est
+ lo-quacity	lo-quac-ity
macroe-co-nomics	macro-eco-nomics
malapropism	mal-a-prop-ism
manuscript	man-u-script
marginal	mar-gin-al
math-e-mati-cian	math-e-ma-ti-cian
mat-tes	mattes
med-i-caid	med-ic-aid
mediocre	medi-ocre
medi-o-crities	medi-oc-ri-ties
me-galith	mega-lith
metabolic	meta-bol-ic
metabolism	me-tab-o-lism
met-a-lan-guage	meta-lan-guage
+ metaphoric	meta-phor-ic
metropo-lis(es)	me-trop-o-lis(es)
metropoli-tan	met-ro-pol-i-tan
mi-croe-co-nomics	micro-eco-nomics
mi-crofiche	mi-cro-fiche
mil-lage	mill-age
milliliter	mil-li-liter
mimeographed	mimeo-graphed
mimeographs	mimeo-graphs
mimi-cries	mim-ic-ries
mi-nis	min-is
min-isym-po-sium(a)	mini-sym-po-sium(a)
min-uter(est)	mi-nut-er(-est)
mis-chievously	mis-chie-vous-ly
mis-ers	mi-sers
mis-ogamy	mi-sog-a-my
mod-elling	mod-el-ling
molecule	mol-e-cule
monar-chs	mon-archs
mon-eylen-der	money-len-der
monochrome	mono-chrome
mo-noen-er-getic	mono-en-er-getic
monoid	mon-oid
monopole	mono-pole
monopoly	mo-nop-oly
monos-pline	mono-spline
monos-trofic	mono-strofic
mono-tonies	mo-not-o-nies
monotonous	mo-not-o-nous
mo-ro-nism	mo-ron-ism
mosquito	mos-qui-to
mu-d-room	mud-room
mul-ti-faceted	mul-ti-fac-eted
mul-ti-pli-ca-ble	mul-ti-plic-able
mul-tiuser	multi-user (better with explicit hyphen)
ne-ofields	neo-fields

	`neon-azi`	neo-nazi
	`newslet-ter`	news-let-ter
	`non-ame`	no-name
	`none-mer-gency`	non-emer-gency
	`nonequiv-ari-ance`	non-equi-vari-ance
+	`nonethe-less`	none-the-less
	`noneu-clidean`	non-euclid-ean
	`non-i-so-mor-phic`	non-iso-mor-phic
	`nonpseu-do-com-pact`	non-pseudo-com-pact
	`non-s-mooth`	non-smooth
	`nonuni-form(ly)`	non-uni-form(-ly)
	`nore-pinephrine`	nor-ep-i-neph-rine
+	`notwith-stand-ing`	not-with-stand-ing
	`nutcracker`	nut-crack-er
	`oer-st-eds`	oer-steds
	`oligopolist`	oli-gop-o-list
	`oligopoly(ies)`	oli-gop-oly(-ies)
	`operand(s)`	op-er-and(s)
	`orangutan`	orang-utan
	`or-thodon-tist`	or-tho-don-tist
	`or-thok-er-a-tol-ogy`	or-tho-ker-a-tol-ogy
	`or-thoni-tro-toluene`	ortho-nitro-toluene (or-tho-ni-tro-tol-u-ene)
	`overview`	over-view
	`ox-i-dic`	ox-id-ic
	`padding`	pad-ding
	`painlessly`	pain-less-ly
	`pal-mate`	palmate
	`parabola`	par-a-bola
	`parabolic`	par-a-bol-ic
	`paraboloid`	pa-rab-o-loid
	`paradigm`	par-a-digm
	`parachute`	para-chute
	`paradimethyl-ben-zene`	para-di-methyl-benzene (para-di-meth-yl-ben-zene)
	`paraflu-o-ro-toluene`	para-fluoro-toluene (para-flu-o-ro-tol-u-ene)
	`para-g-ra-pher`	para-graph-er
	`par-ale-gal`	para-le-gal
	`par-al-lelism`	par-al-lel-ism
	`para-m-ag-netism`	para-mag-net-ism
	`paramedic`	para-medic
	`param-ethy-lanisole`	para-methyl-anisole (para-meth-yl-an-is-ole)
	`parametrize`	pa-ram-e-trize
	`paramil-i-tary`	para-mil-i-tary
	`paramount`	para-mount
	`pathogenic`	path-o-gen-ic
	`pee-vish(ness)`	peev-ish(-ness)
	`pen-tagon`	pen-ta-gon
	`petroleum`	pe-tro-le-um
	`phe-nomenon`	phe-nom-e-non
	`philatelist`	phi-lat-e-list
+	`phoneme`	pho-neme
	`phone-mic`	pho-ne-mic
	`phos-pho-ric`	phos-phor-ic
+	`pho-tographs`	pho-to-graphs
	`pi-cador`	pic-a-dor
	`pi-ran-has`	pi-ra-nhas
	`pla-ca-ble`	placa-ble
	`plea-sance`	pleas-ance
	`poltergeist`	pol-ter-geist
	`polyene`	poly-ene
	`polyethy-lene`	poly-eth-yl-ene
	`polygamist(s)`	po-lyg-a-mist(s)
	`poly-go-niza-tion`	polyg-on-i-za-tion
	`polyphonous`	po-lyph-o-nous
	`polystyrene`	poly-styrene
	`pomegranate`	pome-gran-ate
	`poroe-las-tic`	poro-elas-tic
	`porous`	por-ous
	`postam-ble`	post-am-ble
	`postscript`	post-script
	`pos-tu-ral`	pos-tur-al
	`pream-ble`	pre-am-ble
	`preloaded`	pre-loaded
	`prepar-ing`	pre-par-ing
	`preprint(s)`	pre-print(s)
	`pre-pro-ces-sor`	pre-proces-sor
	`pre-s-plit-ting`	`\pre-split-ting`
	`priestesses`	priest-esses
	`pret-typrinter(ing)`	pret-ty-prin-ter(ing)
	`pro-ce-du-ral`	pro-ce-dur-al
	`pro-cess`	process*
	`procu-rance`	pro-cur-ance
	`pro-ge-nies`	prog-e-nies
	`progeny`	prog-e-ny
	`pro-hibitive(ly)`	pro-hib-i-tive(-ly)
	`prosci-utto`	pro-sciut-to
	`protester(s)`	pro-test-er(s)
	`protestor(s)`	pro-tes-tor(s)
+	`pro-tolan-guage`	pro-to-lan-guage
	`pro-to-ty-pal`	pro-to-typ-al
	`pseu-dod-if-fer-en-tial`	pseu-do-dif-fer-en-tial
	`pseud-ofi-nite`	pseu-do-fi-nite
	`pseud-ofinitely`	pseu-do-fi-nite-ly
	`pseud-o-forces`	pseu-do-forces
	`pseudonym`	pseu-do-nym
	`pseu-doword`	pseu-do-word
	`psychedelic`	psy-che-del-ic
	`psy-chs`	psychs
	`pubescence`	pu-bes-cence
	`quadding`	quad-ding
	`quadratic(s)`	qua-drat-ic(s)
	`quadra-ture`	quad-ra-ture
	`quadriplegic`	quad-ri-pleg-ic
	`quain-ter(est)`	quaint-er(-est)
	`quasiequiv-a-lence`	qua-si-equiv-a-lence or quasi-
	`quasi-hy-ponor-mal`	qua-si-hy-po-nor-mal
	`quasir-ad-i-cal`	qua-si-rad-i-cal
	`quasiresid-ual`	qua-si-resid-ual
	`qua-sis-mooth`	qua-si-smooth
	`qua-sis-ta-tion-ary`	qua-si-sta-tion-ary
	`qu-a-sito-pos`	qua-si-topos
	`qu-a-si-tri-an-gu-lar`	qua-si-tri-an-gu-lar
	`quintessence`	quin-tes-sence
	`quintessen-tial`	quin-tes-sen-tial
	`rab-bi-try`	rab-bit-ry
	`ra-dio-g-ra-phy`	ra-di-og-ra-phy
	`raf-f-ish(ly)`	raff-ish(-ly)
	`ramshackle`	ram-shackle
	`ravenous`	rav-en-ous
	`re-ar-range-ment`	re-arrange-ment
	`re-ciproc-i-ties`	rec-i-proc-i-ties
	`reci-procity`	rec-i-proc-i-ty
	`rect-an-gle`	rec-tan-gle
	`ree-cho`	re-echo

reprint(s)	re-print(s)
restorable	re-stor-able
retrofit(ted)	retro-fit(-ted)
rhinoceros	rhi-noc-er-os
righ-teous(ness)	right-eous(-ness)
ringleader	ring-leader
robot	ro-bot
robotics	ro-bot-ics
roundtable	round-table
salesclerk	sales-clerk
salescle-rks	sales-clerks
saleswoman(en)	sales-woman(en)
salmonella	sal-mo-nel-la
sarsaparilla	sar-sa-par-il-la
sauerkraut	sauer-kraut
sca-to-log-i-cal	scat-o-log-i-cal
schedul-ing	sched-ul-ing
schizophrenic	schiz-o-phrenic
schnauzer	schnau-zer
schoolchild(ren)	school-child(-ren)
schoolteacher	school-teacher
scy-thing	scyth-ing
sec-re-tariat	sec-re-tar-iat
semaphore	sem-a-phore
semester	se-mes-ter
semidef-i-nite	semi-def-i-nite
semi-ho-mo-th-etic	semi-ho-mo-thet-ic
semir-ing	semi-ring
semiskilled	semi-skilled
seroepi-demi-o-log-i-cal	sero-epi-de-mi-o-log-i-cal
ser-vomech-a-nism	ser-vo-mech-anism
setup	set-up
severely	se-vere-ly
sha-peable	shape-able
shoestring	shoe-string
sidestep	side-step
sideswipe	side-swipe
skyscraper	sky-scraper
smokestack	smoke-stack
snorke-l-ing	snor-kel-ing
solenoid	so-le-noid
so-lute(s)	solute(s)
sovereign	sov-er-eign
spaces	spa-ces
specious	spe-cious
spelunker	spe-lunk-er
spendthrift	spend-thrift
spheroid(al)	spher-oid(-al)
sph-inges	sphin-ges
spi-cily	spic-i-ly
spinors	spin-ors
spokeswoman(en)	spokes-woman(en)
sportscast	sports-cast
sportively	spor-tive-ly
sportswear	sports-wear
sportswriter	sports-writer
sprightlier	spright-lier
squeamish	squea-mish
stan-dalone	stand-alone
startling(ly)	star-tling(-ly)
statis-tics	sta-tis-tics
stealthily	stealth-ily
steeplechase	steeple-chase
stochas-tic	sto-chas-tic
strangeness	strange-ness
stratagem	strat-a-gem
stretchier	stretch-i-er
stronghold	strong-hold
strongest	strong-est
stupi-der(est)	stu-pid-er(-est)
summable	sum-ma-ble
su-perego	super-ego
su-pere-gos	super-egos
supremacist	su-prema-cist
surveil-lance	sur-veil-lance
swim-m-ingly	swim-ming-ly
symp-tomatic	symp-to-matic
syn-chromesh	syn-chro-mesh
syn-chronous	syn-chro-nous
syn-chrotron	syn-chro-tron
talkative	talk-a-tive
tapestry(ies)	ta-pes-try(ies)
tarpaulin	tar-pau-lin
tele-g-ra-pher	te-leg-ra-pher
telekinetic	tele-ki-net-ic
teler-obotics	tele-ro-bot-ics
tem-porar-ily	tem-po-rar-ily
tenure	ten-ure
testbed	test-bed
tex-twidth	text-width
tha-la-mus	thal-a-mus
ther-moe-las-tic	ther-mo-elas-tic
times-tamp	time-stamp
toolkit	tool-kit
to-po-graph-i-cal	topo-graph-i-cal
to-ques	toques
traitorous	trai-tor-ous
transceiver	trans-ceiver
transgress	trans-gress
transver-sal(s)	trans-ver-sal(s)
transvestite	trans-ves-tite
traversable	tra-vers-a-ble
traver-sal(s)	tra-ver-sal(s)
treacheries	treach-eries
troubadour	trou-ba-dour
turkey	tur-key
turnaround	turn-around
ty-pal	typ-al
unattached	un-at-tached
unerringly	un-err-ing-ly
un-friendly(ier)	un-friend-ly(i-er)
va-guer	vaguer
vaudeville	vaude-ville
vi-cars	vic-ars
vil-lai-ness	vil-lain-ess
viviparous	vi-vip-a-rous
voiceprint	voice-print
vs-pace	\vspace
wadding	wad-ding
wallflower	wall-flower
wastew-a-ter	waste-water
waveg-uide	wave-guide
wavelets	wave-lets
we-b-like	web-like
weeknight	week-night
wheelchair	wheel-chair
whichever	which-ever
whitesided	white-sided

`whites-pace`	white-space
`widespread`	wide-spread
`wingspread`	wing-spread
`witchcraft`	witch-craft
`wordspac-ing`	word-spac-ing
`workhorse`	work-horse
`wraparound`	wrap-around
`wretched(ly)`	wretch-ed(-ly)
`yesteryear`	yes-ter-year

Names and non-English words used in English text

`al-ge-brais-che`	al-ge-brai-sche
`Al-legheny`	Al-le-ghe-ny
`Arkansas`	Ar-kan-sas
`Aus-tralasian`	Aus-tral-asian
`au-toma-tisierter`	auto-mati-sier-ter
`Be-di-enung`	Be-die-nung
`bib-li-ographis-che`	bib-li-o-gra-phi-sche
`Boston`	Bos-ton
`Brow-n-ian`	Brown-ian
`Brunswick`	Bruns-wick
`Bu-dapest`	Bu-da-pest
`Caribbean`	Car-ib-bean
`Charleston`	Charles-ton
`Char-lottesville`	Char-lottes-ville
`Columbia`	Co-lum-bia
`Czechoslo-vakia`	Czecho-slo-va-kia
`Di-jk-stra`	Dijk-stra
`dy-namis-che`	dy-na-mi-sche
`En-glish`	Eng-lish
`Eu-le-rian`	Euler-ian
`Evanston`	Evan-ston
`Febru-ary`	Feb-ru-ary
`Festschrift`	Fest-schrift
`Florida`	Flor-i-da
`Forschungsin-sti-tut`	For-schungs-in-sti-tut
`funk-t-sional`	funk-tsional
`Gaus-sian`	Gauss-ian
`Greif-swald`	Greifs-wald
`Grothendieck`	Grothen-dieck
`Grundlehren`	Grund-leh-ren
`Haifa`	Hai-fa
`Hamil-to-nian`	Hamil-ton-ian
`Helsinki`	Hel-sinki
`Her-mi-tian`	Her-mit-ian
`Hi-bbs`	Hibbs
`Hokkaido`	Hok-kai-do
`Jan-uary`	Jan-u-ary
`Japanese`	Ja-pa-nese
`Kadomt-sev`	Kad-om-tsev
`Kansas`	Kan-sas
`Karl-sruhe`	Karls-ruhe
`Ko-rteweg`	Kor-te-weg
`Lan-caster`	Lan-cas-ter
`Leg-en-dre`	Le-gendre
`Le-ices-ter`	Leices-ter
`Lip-s-chitz(ian)`	Lip-schitz(-ian)
`Louisiana`	Lou-i-si-ana
`Manch-ester`	Man-ches-ter
`Marko-vian`	Mar-kov-ian
`Mas-sachusetts`	Mass-a-chu-setts
`Min-neapo-lis`	Min-ne-ap-o-lis
`Min-nesota`	Min-ne-sota
`Moscow`	Mos-cow
`Nachrichten`	Nach-richten
`Nashville`	Nash-ville
`Ni-jmegen`	Nij-me-gen
`Noethe-rian`	Noe-ther-ian
`No-ord-wi-jk-er-hout`	Noord-wijker-hout
`Novem-ber`	No-vem-ber
`Palermo`	Pa-ler-mo
`Philadel-phia`	Phil-a-del-phia
`Poincare`	Poin-care
`Po-ten-tial-gle-ichung`	Po-ten-tial-glei-chung
`rathskeller`	raths-kel-ler
`Rie-man-nian`	Rie-mann-ian
`Ry-d-berg`	Ryd-berg
`schot-tis-che`	schot-tische
`Schrodinger`	Schro-ding-er
`Schwabacher`	Schwa-ba-cher
`Schwarzschild`	Schwarz-schild
`Septem-ber`	Sep-tem-ber
`Stokess-che`	Stokes-sche
`Stuttgart`	Stutt-gart
`Susque-hanna`	Sus-que-han-na
+ `Taube-rian`	Tau-ber-ian
`tech-nis-che`	tech-ni-sche
`Ten-nessee`	Ten-nes-see
+ `ty-pographique`	ty-po-graphique
`Ukrainian`	Ukrain-ian
`ve-r-all-ge-mein-erte`	ver-all-ge-mein-erte
`Vere-ini-gung`	Ver-ein-i-gung
`Verteilun-gen`	Ver-tei-lun-gen
`Wahrschein-lichkeit-s-the-o-rie`	Wahr-schein-lich-keits-theo-rie
`Werthe-rian`	Wer-ther-ian
`Winch-ester`	Win-ches-ter
`Yingy-ong Shuxue Jisuan`	Ying-yong Shu-xue Ji-suan
`Zealand`	Zea-land
`Zeitschrift`	Zeit-schrift

◇ Barbara Beeton

LaTeX News

Issue 10, December 1998

Five years of LaTeX 2ε

Since this is the 10th edition of LaTeX News, the (no longer) New Standard LaTeX must have hit the streets almost this long ago. In fact it was only the beta-version that some people got just in time for Christmas 1993, and since then there has been a lot of tidying-up and smoothing of rough edges (not to mention a few bug fixes!).

Maybe it is time for something more radically different to emerge and be hungrily adopted by the world; but don't panic, we shall be maintaining what you have now for a long time yet. Amongst the more polite things that have been written about our efforts, we found that this quote (somewhat censored to protect the guilty) well reflects some of our feelings about working on LaTeX over the years: *the mere existence of LaTeX 2ε is a great miracle.*

Restructuring the LaTeX distribution

Since the (once) 'new' standard LaTeX has reached such a venerable age, we are reviewing the way in which the system is presented to the world.

An early intention is to define, given the wide variety of good packages now available, what now constitutes a useful installation of LaTeX. We also hope that such a definition will help document portability if it leads to a future in which a LaTeX class designer can reasonably assume that a known list of facilities will be there for all users (so that each class need not supply them).

As a first small step towards this definition, we shall replace the `latex/packages` subdirectory on CTAN. This directory was a curious mixture of the important, such as the LaTeX `tools`, that any self-respecting LaTeX installation ought to have, and the esoteric or experimental.

The esoterica from `packages` will be moved to new locations, as follows:

`expl3` to `latex/exptl/project`
`mfnfss` to `latex/contrib/supported/mfnfss`

The subdirectory that replaces `packages` will be called `latex/required`; all the other sub-directories of `packages` will be moved there.

LaTeX Project on the Internet

A new `latex-project.org` domain has been registered. The web site is not yet fully functional but the old LaTeX pages from CTAN are available at `http://www.latex-project.org/` and the LaTeX bug reporting address has been changed to `latex-bugs@latex-project.org`.

Restructuring the LaTeX package licenses

Several people have requested an easy mechanism for the distribution of LaTeX packages and other software "under the same conditions as LaTeX". The old `legal.txt` file was unsuitable as a general licence as it referred to specific LaTeX authors, and to specific files.

Therefore, in this release `legal.txt` contains just the copyright notice and a reference to the new *LaTeX Project Public License* (LPPL) for the distribution and modification conditions. The `tools`, `graphics`, and `mfnfss` packages also now refer to this license in their distribution notices.

Support for Cyrillic encodings

Basic Cyrillic support, as announced in LaTeX News 9, is now finally an official part of LaTeX. It includes support for the following standard Cyrillic font encodings (this list may grow): `T2A T2B T2C X2`.

It also includes various Cyrillic input encodings (20 in total, including commonly used variants and Mongolian Cyrillic encodings). This provides platform independent and sophisticated basic support for high-quality typesetting in various Cyrillic-based languages.

For further information see the file `cyrguide.tex`.

Tools distribution

The `varioref` package has been extended to support textual page references to a range of objects: e.g., if `eq-first` and `eq-last` are the label names for the first and last equation in a sequence, then you can now write

```
see~\vrefrange{eq-first}{eq-last}
```

This results in different text depending on whether both labels fall on the same page.

Some additional user commands, as well as building-blocks for writing private extensions, are described in the accompanying documentation.

The Treasure Chest

Package tours from CTAN

> Give us the tools, and we will finish the job.
>
> Churchill, in a broadcast dated February 1941

Well, everyone's a Churchill these days — we all want good tools to help, not hinder, us as we get on with our jobs, albeit more peaceful ones than in 1941. Hopefully some of these packages will provide just the tools *you* need.

The selection focuses on tools to achieve certain layout effects for authors:[1] itemized lists shaped as paragraphs; managing the use and definition of acronyms; epigraphs — those 'pithy quotations often found at the start (or end) of a chapter' (or the start of a column, in this case); and two types of 'hanging' — paragraphs and punctuation. So go fetch the files and have some fun!

All packages described in this column are available in the following location on CTAN:[2]

`macros/latex/contrib/supported`

Only the specific subdirectory will be identified in individual descriptions, to reduce the repetition — and the line length of such paths!

Another commonality: many of these packages use two mechanisms afforded by LaTeX 2$_\varepsilon$ — `.ins` and `.dtx` files.

The `.ins` file is run through LaTeX in order to generate the style file; the `.dtx` file is usually the main source of documentation and should therefore also be processed. Different TeX installations store such files differently; check your documentation or your machine on where other such files are located, and add these to the same places.

Note that, on occasion, it may be necessary to process the documentation file two or even three times, in order to resolve all cross-references. Processing of these files sometimes generates additional files which the authors have found useful to include, along with the usual `.log` and `.aux` files.

1 `paralist`

1.1 Quick Tour

Package: `paralist`
This is version 1.8, dated 29 Apr. 1999.

Author: Bernd Schandl
`bschand@math.clemson.edu`

Compatible with: LaTeX 2$_\varepsilon$

Keywords:
in-paragraph list environments, customizable labels

Purpose:
`paralist.sty` provides some new list environments. Itemized and enumerated lists can be typeset within paragraphs, as paragraphs and in a compact version. Most environments have optional arguments to format the labels. Additionally, the default LaTeX environments `itemize` and `enumerate` can be extended to use a similar optional argument.

Explanation of the name: "lists in paragraph shape."

Location on CTAN: `/paralist`

Files to fetch:
`README`
`paralist.ins`
`paralist.dtx`
`license.txt`.[3]

How to install:
Put the `.dtx` and `.ins` files with the other class and style files on your system. The `README` file provides processing information.

Files generated:
`paralist.sty` (the style file)
`paralist.dvi` (documentation)
`paralist.drv` (more detailed documentation with code + index)

Documentation:
Processing the `.dtx` file yields 7 pages. Additional documentation can also be generated, following instructions in the `README` file.

Table of Contents:

1. Introduction
2. Package Options
3. Formatting the Labels
4. Defaults for Labels and Margins

[1] I would like to thank the authors — Bernd Schandl, Tobias Oetiker, and Peter Wilson — who were so cooperative and helpful as I worked through their package instructions, the cumulative package interferences, and so on. It's been fun putting this column together — and it's been a pleasure working with these authors.

[2] Either via the web at `www.tug.org` (follow the links) or via ftp to `ctan.tug.org/tex-archive`.

[3] The new license text (LPPL): "This package can be redistributed and/or modified under the terms of the LaTeX Project Public License distributed from CTAN archives in the directory `macros/latex/base/lppl.txt`; either version 1 of the license, or (at your option) any later version." The regular user can just delete the file after downloading, because a pointer to the license is included in every file anyway.

5. New Environments
   (a) Enumerated
   (b) Itemized
   (c) Descriptive
   (d) Blank
6. Nesting Environments
7. Fine-tuning
8. Bugs and Wishes
9. Acknowledgments

1.2 Detailed tour

The package grew out of `theapa`,[4] and also owes a great deal to David Carlisle's `enumerate` package, as noted by the author.

The documentation details four package options:

1. Two of these allow the redefinition of the standard environments `enumerate` and `itemize`, so that they have an optional argument to format the labels.

2. Another option addresses the space required for labels.

3. The final option is to define some list environments without labels, an odd but sometimes useful option.

The above demonstrates one of the two new environments: **(1)** `asparaenum`, whereby each `\item` is set as a separate paragraph, and **(2)** `inparaenum`, which sets its `\item`s into one continuous paragraph. This paragraph, by the way, demonstrates the use of `inparaenum`, with the addition of a label option, `[\bfseries (1)]`, as opposed to the default LaTeX strategy, which requires customized options for each instance of `\item`.

There are matching environments for the 'itemize' type of list: `asparaitem` and `inparaitem`, also with options to modify their labels. And, as mentioned in the 'Purpose' section, there are compact versions of these environments — where all the vertical skips have been set to zero. It was used above, to set the Table of Contents list.

1.3 Final notes

Here follows a whimsical example of resetting the labels for the `compactitem` environment. Note that since there is no 'counting' going on, `compactenum` would not work.[5]

```
\defaultitem{$\clubsuit$}
            {$\diamondsuit$}
            {$\heartsuit$}
            {$\spadesuit$}
```

- ♣ something first
  - ♢ something first
    - ♡ something first
      - ♠ something first
      - ♠ something second
    - ♡ something second
  - ♢ something second
- ♣ something second

The author has one query (p. 5 in the documentation):

> … does anybody know why `description` has to be defined by the document class and is not defined in `ltlists.dtx`?

Send your replies directly to Bernd, and hopefully he can come back with an answer!

The author also has two comments on additional features to look for:

1. Using `\defaultenum` to redefine the counter styles throughout the whole document is pretty cool, but maybe I am a little biased here.
2. With a little extra work (by redefining `\pl@hook`) it is possible to use basically everything (footnote symbols, Greek characters …) as counters for enumerated lists.

The flexibility the package affords, while taking a few tries to sort out, cannot fail to intrigue and please writers of long documents — or rather, of multi-part paragraphs! Lawyers come to mind, for some reason `;-)`

2 `acronym`

2.1 Quick Tour

Package: `acronym`
This is version 1.3, dated 1996/09/19.

Author: Tobias Oetiker
`oetiker@ee.ethz.ch`

Compatible with: LaTeX 2_ε

Keywords:
acronyms, mottos, topical quotations; full and short versions; auto-generation of acronym listing

Purpose:
The `acronym` package ensures that all acronyms

[4] A LaTeX/AMS-LaTeX document style option (supported) for use with the `theapa` BibTeX bibliography style (`theapa.bst`), by Young U. Ryu `ryoung@utdallas.edu`. Latest version is v2.5.1 (May 1992). See also `theapa.tex`, `theapa.bst`.

[5] Actually, I originally had used `compactenum`, not thinking about this counting being the essential difference between the two environments. So now that I've made that mistake for you … !

used in the text are spelled out in full at least once. The package provides several commands and one environment for dealing with acronyms.

Location on CTAN: `/acronym`

Files to fetch:

`acronym.ins`
`acronym.dtx`

How to install:

Process both the `.ins` and `.dtx` files and then store where other such files are kept.

Files generated:

`acronym.sty` (the style file)
`acronym.dvi` (documentation)
`acrotest.tex` (sample file)

Documentation:

The `.dtx` documentation runs to 5 pages; no separate `readme` is provided. Instructions on processing the `.dtx` and test file are in the `.ins` file, but no explicit instructions that the `.ins` file be run through LaTeX 2_ε are provided.

Table of Contents:

2.2 The detailed tour

This package covers such a small corner of the writing/editing/typesetting and yet, it will probably become invaluable for those whose writings require that they manage and control large numbers of acronyms.

The writing convention of using the complete name or title ('Full Name'), followed by the acronym in parentheses, for first appearance, and then only the acronym ('short name') for subsequent instances, is handled via four macros for the various treatments.

In addition, should one want to print up the acronyms in an appendix or other section, with both full and short names, along with any additional description, an `acronym` environment is available.

The documentation provides the verbatim input code for the sample file, to demonstrate the coding. A final section, 'The implementation', provides details on how the macros work.

2.3 Sample file

In the early nineties, GSM was deployed in many European countries. Global System for Mobile communication (GSM) offered for the first time international roaming for mobile subscribers. The GSM's use of Time Division Multiple Access (TDMA) as its communication standard was debated at length. And every now and then there are big discussions whether Code Division Multiple Access (CDMA) should have been chosen over TDMA.

If you want to know more about Global System for Mobile communication (GSM), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA) and other acronyms (oa), just read a book about mobile communication.

2.3.1 Acronyms Used

GSM Global System for Mobile communication. GSM is the new standard for digital cellular communication in Europe.

TDMA Time Division Multiple Access. Some would say that this is not as good as CDMA.

CDMA Code Division Multiple Access. The spread spectrum modulation used in the Qualcomm system.

2.4 Final notes

Incorporating the `acronym` package into *TUGboat*'s collection of macros has led to a clash in control sequence names: we both use `\acro`! In *TUGboat*, this is used to set ALLCAPS acronyms into the next smaller font: ALLCAPS. However, it's also a very logical macro to use for acronym macros. Our solution? After loading the *TUGboat* macros, but before loading the `acronym` package, the following lines were inserted in the preamble:

```
\let\TBacro\acro
\let\acro\relax
```

And then all instances of `\acro` for ALLCAPS were changed to as `\TBacro`.

One more point. As any editor knows, the first thing we do is go in and change things! Well, there is one small part which can't be changed — yet. In testing, I found that the first argument for `\acro`, where the initials are put, won't accept changes in font size or style. The author has agreed that this would be a worthwhile feature and will incorporate it in the next upgrade. We will begin to carry upgrade information as it becomes available for this, and any other package found in the 'Treasure Chest'.

3 `epigraph`

3.1 Quick Tour

Package: `epigraph`

This is version 1.1, dated 1998/11/29.

Author: Peter R. Wilson
`peter.r.wilson@boeing.com`
Compatible with: LaTeX 2_ε
Keywords:
epigraphs, epigram, quotation
Purpose:
The `epigraph` package is designed for typesetting epigraphs — pithy quotations often found at the start (or end) of a chapter. It provides commands for typesetting a single epigraph and environments for typesetting a list of epigraphs. Epigraphs can be typeset at either the left, the center, or the right of the typeblock. Similarly, the source line can appear in different positions.
Location on CTAN: `/epigraph`
Files to fetch:
`README`
`epigraph.ins`
`epigraph.dtx`
How to install:
The `README` file provide instructions for two levels of detail. The usual sequence of processing the `.ins` file and then the `.dtx` file (run this latter three times, to resolve all cross-references) yield the style file and documentation. Store these where other such files are kept.

For additional processing of an index `.idx` file, there are instructions for using `makeindex`. But it is not necessary to do the second level in order to begin installing and using the package.
Files generated:
`epigraph.sty` (the style file)
`epigraph.dvi` (documentation)
`epigraph.toc` and `epigraph.idx` (for `makeindex` documentation)
Documentation:
The documentation runs to 8 pages, with the macros nicely set off in the left margin, as is common with `.dtx` material.

Table of Contents:

1. Contents
2. Introduction
3. The `epigraph` package
   (a) The `epigraph` command
   (b) The `epigraphs` environment
   (c) Epigraphs before chapter headings
4. General
5. The package code
   (a) Epigraphs before a chapter title
6. References

3.2 The detailed tour

The documentation seems to cover almost all possible combinations and situations one would normally expect to run into — however, see the first epigraph, below, to explain the unanticipated uses!

The macros are very simple, the explanations clear, with sufficient verbatim samples. I like it!

Below are the epigraphs used in last year's issues of *TUGboat*:

> **Book design no less than ship design or any other kind of design is a matter of looking backward and forward at the same time. The grandeur of a volume's illustrations may so dominate its design that portability and readability become subsumed in the dominant objective.**
>
> **Petroski (1997), in *TUGboat*, 19,1**

> **No matter how many palettes of buttons and how many menu options are offered, users of a program will always want to do something the author has not foreseen. Adding still more buttons and menus is not the answer.**
>
> **Hayes (1995), in *TUGboat*, 19,2**

> **[proceedings issues use this space to provide details on the conference]**
>
> ***TUGboat*, 19,3**

> **Certainly, if I were a publishing house, if I were in the publishing business myself, I would have probably had ten different versions of TeX by now for ten different complicated projects that had come in. They would all look almost the same as TeX, but no one else would have this program — they wouldn't need it, they're not doing exactly the book that my publishing house was doing.**
>
> **Knuth (1996), in *TUGboat*, 19,4**

I added `\itemsep=1.5pc` to get some space between each one. Also, by setting the following parameter (line-wrapped to fit the column), the font size/style can be changed for the entire environment:

```
\renewcommand{\epigraphsize}%
            {\bfseries\footnotesize}
```

The documentation doesn't mention the bit about font style being changeable via this command, but it does work. Also, each of the two arguments — one for the text, one for the source — will accept the usual LaTeX font commands.

3.3 Final notes — and a query

The author, Peter Wilson, offered up the following epigraphs for the opening of this issue of the Treasure Chest. And while I like my own little discovery, his are too good to leave out. Notice the variations in position for the source line(s).

> The epigraph is among the most delightful of scholarly habits.
>
> ---
>
> Mary-Claire Van Leunen,
> *A Handbook for Scholars*, 1978

> I hate quotations. Tell me what you know.
>
> ---
>
> Ralph Waldo Emerson, *Journals*, May 1849

> QUOTATION, n. The act of repeating erroneously the words of another. The words erroneously repeated.
>
> ---
>
> Ambrose Bierce,
> **The Devil's Dictionary**, 1906

> There is no amount of praise which a man and an author cannot bear with equinimity. Some authors can even stand flattery.
>
> ---
>
> *Maurice Baring*, Dead Letters, 1910

The query is from the author:

> As a result of someone's posting to the newsgroup `comp.text.tex` I am considering adding a facility in the `epigraph` package for putting typeset material on the (verso) page immediately before the (recto) page on which the chapter heading is typeset. *If* I do this (and it would be helpful to know if this was a common need), then it would be in a new version of the package.

4 `hanging`

4.1 Quick Tour

Package: `hanging`
This is version 1.0, dated 1998/11/29.

Author: Peter R. Wilson
`peter.r.wilson@boeing.com`

Compatible with: LaTeX 2$_\varepsilon$

Keywords:
hanging paragraphs, hanging punctuation

Purpose:
The `hanging` package provides facilities for defining hanging paragraphs and hanging punctuation.

Location on CTAN: `/hanging`

Files to fetch:
`README`
`hanging.ins`
`hanging.dtx`

How to install:
The `README` file provides the by-now familiar instructions on handling the two other files.

Files generated:
`hanging.sty` (the style file)
`hanging.dvi` (documentation)
`hanging.toc` and `hanging.idx` (for `makeindex` documentation)

Documentation:
Documentation is very clear and spare: 5 pages in all, including the documented source code. The `README` file provides processing information.

Table of Contents:

1. Contents
2. Introduction
3. The `hanging` package
   (a) Hanging paragraphs
   (b) Hanging punctuation
4. The package code
   (a) Hanging paragraphs
   (b) Hanging punctuation
5. References

4.2 The detailed tour

Very nifty little set of macros! Very simple to understand and then use. For hanging paragraphs, you can choose between one macro for one paragraph, or an environment for several paragraphs. Both commands take two arguments: one to specify the left indent, the second to specify how many lines to print before starting the hanging indent. For example,

```
\hangpara{3pc}{1}
```

yields a 3pc indent from the left, and only one full line of text before the indent comes into play. Now, by setting that first argument to a negative value (`-3pc`), the indent switches from the left margin to the right, as done with this paragraph. The potential for playing around should be obvious!

The `hangparas` environment can be contrasted with the effects of LaTeX's `description`, mainly in terms of ease of control of the `\leftmargin` value. Below are the same two texts, first done with `hangparas` (with a 1pc indent), then with `description`.

paralist: provides some new list environments. Itemized and enumerated lists can be typeset within paragraphs, as paragraphs and in a compact version. Most environments have optional arguments to format the labels.

acronym: ensures that all acronyms used in the text are spelled out in full at least once. The package provides several commands and one environment for dealing with acronyms.

epigraph: is designed for typesetting epigraphs — the pithy quotations often found at the start (or end) of a chapter. It provides commands for typesetting a single epigraph and environments for typesetting a list of epigraphs. Epigraphs can be typeset at either the left, the center, or the right of the typeblock.

hanging: provides facilities for defining hanging paragraphs and hanging punctuation.

paralist: provides some new list environments. Itemized and enumerated lists can be typeset within paragraphs, as paragraphs and in a compact version. Most environments have optional arguments to format the labels.

acronym: ensures that all acronyms used in the text are spelled out in full at least once. The package provides several commands and one environment for dealing with acronyms.

epigraph: is designed for typesetting epigraphs — the pithy quotations often found at the start (or end) of a chapter. It provides commands for typesetting a single epigraph and environments for typesetting a list of epigraphs. Epigraphs can be typeset at either the left, the center, or the right of the typeblock.

hanging: provides facilities for defining hanging paragraphs and hanging punctuation.

As for hanging punctuation (look carefully at the margins of this paragraph), while the author views it as an interesting — if less-than-practical — facility, it does work! There is a `hangpunct` environment, within which no special coding of the following punctuation marks is required: `‘ ’ . ! ? : ; , .` Where the characters `‘ ’ .` are required in an ‘unhung’ version, alternate macros are provided.

4.3 Final notes

In working with so many packages in one file, there are bound to be some small interference hiccups. One has already been mentioned: the sharing of the `\acro` command (see Section 2).

Another has arisen with `hanging`: a command not sufficiently localized provoked the following error message:

```
LaTeX Error: Command \allowhyphens
already defined.
```

The quick solution offered by the author:

> Because of the time constraints [to get this issue out], edit the generated **`hanging.sty`** file and change each occurrence of **`\allowhyphens`** to **`h@ngallowhyphens`**. It is an internal command in the package, and to be consistent I should have called it **`\h@ngallowhyphens`** in the first place.
>
> It is fortunate that the **`\allowhyphens`** command was specified using **`\newcommand`**, as advised for package writers, rather than using the **`\def`** command. In the latter case the original definitions would have been silently clobbered and probably cause the *TUGboat* editors much undeserved grief. But as it was, warnings were given of the problem so that a more robust solution could be developed.

That is, the *TUGboat* command is intended for users, so it should *not* be restricted. That's what we've done, and it did the trick. When the next upgrade comes on-line, it will be reported in this column.

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

Helpful hint for finding files on the CTAN archives via ftp

Don't know where to find the package you want? The following shows how to use the `quote site index` command, to quickly locate packages. Our example is for `paralist`.

```
ftp> quote site index paralist
200-index paralist
200-NOTE. This index shows at most 20 lines. for a full list of files,
200-retrieve /tex-archive/FILES.byname
200-1999/04/01 |    2456 | macros/latex/contrib/supported/paralist/README
200-1999/04/01 |     620 | macros/latex/contrib/supported/paralist/license.txt
200-1999/04/01 |   37787 | macros/latex/contrib/supported/paralist/paralist.dtx
200-1999/04/01 |    1092 | macros/latex/contrib/supported/paralist/paralist.ins
200  (end of 'index paralist')
```

Abstracts

Die TEXnische Komödie Contents of Some Past Issues

9. Jahrgang, Heft 1/1997 (Mai 1997)

Luzia DIETSCHE, [Editorial]; p. 3

A short statement commenting on the current issue. In particular, she points out that articles on internal matters of DANTE occupy most of the pages. Moreover, she is no longer a member of the presidium, and has not written the protocol, but she is the editor responsible for the report of the technical council.

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

Joachim LAMMARSCH, Grußwort
[Welcome message]; pp. 4–7

A short comment on club matters by the president of DANTE, mainly describing recent changes to the presidium of DANTE, and other organizational matters. Moreover, the case of unfounded bank transfers initiated by AOL is still not resolved.

[Ehrenfried JUST],
Protokoll der 16. Mitgliederversammlung von DANTE, Deutschsprachige Anwendervereinigung TEX e.V. [Protocol of the 16th assembly of members of DANTE]; pp. 7–18

This is the official report of the members' meeting held in München (February 27, 1997). As usual, it is concerned with various formal and organizational matters, and reports on the situation of DANTE. Among the topics treated are the funding of NTS (DM 30,000.–), the decreasing membership numbers, the DANTE-CTAN-CDROM, and the case of AOL. The general assembly appoints four new honorary members: Donald E. Knuth, Barbara Beeton, Eberhard Mattes, and Luzia Dietsche. (The other honorary members are Peter Sandner, Klaus Thull, and Yannis Haralambous.) Finally, a new presidium is elected: Joachim Lammarsch and Friedhelm Sowa are reelected, while Uwe Untermarzoner and Luzia Dietsche do not stand for office again. They are replaced by Marion Neubauer and Günter Partosch.

Marion NEUBAUER, Neue Software bei DANTE e.V. [New software from DANTE e.V.]; pp. 19–20

A short description of the software packages distributed by DANTE.

○ *TEX-Theatertage*
[TEX theatre festival]; pp. 21–24:

Martin SCHRÖDER, DANTE '97 in München
[DANTE '97 in Munich]; pp. 21–24

A personal account of DANTE '97 (February 26–28, 1997) in Munich, describing briefly the proceedings of the conference.

○ *Bretter, die die Welt bedeuten*
[The stage is the world]; pp. 25–44:

Marion NEUBAUER, Feinheiten bei wissenschaftlichen Publikationen – Mikrotypographie-Regeln, Teil II [Fine points of scientific publishing— microtypographic rules, part II]; pp. 25–44

This is the second part of a (long and detailed) article which attempts a survey of the rules for German text (with some additional rules for foreign languages) and how they can be followed in LATEX. This part deals with numbers and digits, braces, spaces, hyphenations (and overfull boxes). A bibliography concludes the article. (For the first part see Vol. 8, issue 4.)

○ *Was Sie schon immer über TEX wissen wollten ...* [What you always wanted to know about TEX ...]; p. 45:

Bernd RAICHLE, Abkürzungspunkt am Satzende
[Abbreviation point at the end of a sentence]; p. 45

`xpoint.sty`, a LATEX-package by Bernd Raichle (similar to `xspace`), allows macros for abbreviations to end with a point which is suppressed if it occurs at the end of a sentence.

○ *TEX-Beiprogramm* [TEX co-features]; pp. 46–63:

Dieter GLÖTZEL, Database-Publishing mit Oberon und LATEX [Database publishing with Oberon and LATEX]; pp. 46–54

This case study describes the (successful) production of an official directory for the (new) German ZIP codes using Oberon (as programming language) and TEX (as typesetting engine). Finally, the author pleads for the use of public domain software in commercial environments: *Freeware is better software!*

Bericht des technischen Beirats
[Report of the technical council]; pp. 55–63

A collection of short reports (by the appointed coordinators) on the various hardware platforms and other special topics. In particular, eTEX, version 1.1, the `ec`-fonts, version 1.0, and the (new) mailbox of DANTE are presented.

○ *Leserbriefe* [Letters]; p. 64:

Walter SCHMIDT, Eine TEX-Umgebung für OS/2, Heft 4/96 [A TEX environment for OS/2, issue 4/96]; p. 64

The author comments that using the standard edition of EPMTEX is preferable since it is easier to install than the pro(fessional) version described in the cited article, but fully sufficient.

○ *Rezensionen* [Reviews]; pp. 65–68:

Thomas KOCH, *Der LATEX-Wegweiser*
[The LATEX signpost]; pp. 65–66

A review of a book (by Christine Detig) with this title (International Thomson Publishing, Bonn 1997). It is a quick introduction to LATEX2$_\varepsilon$, written in question-and-answer style and using a concrete sample document. The book's message (according to the review): *LATEX is not complicated!*

Andreas SCHLECHTE, *Das TEXikon – Referenzhandbuch für TEX and LATEX* [The TEXikon — Reference handbook for TEX and LATEX]; pp. 66–68

A review of a book (by Stefan Schwarz and Rudolf Potuček) with this title (Addison-Wesley, Bonn 1997). The book is a lexicon of TEX and LATEX commands (in alphabetical order, with detailed description). The bottom line: A useful reference. However, possible side effects of mixing LATEX with (plain) TEX commands are not sufficiently treated. In order to illustrate this, the reviewer presents an example (by Bernd Raichle), a short piece of (LATEX) code, and asks the reader to predict (and explain) its result. (The solution is promised for the next issue.)

○ *Spielplan* [Repertory]; pp. 69–71:

The international and national calendar, including the announcement of *TUG'97 – TEX Comes Home* in San Francisco (July 28 – August 1, 1997).

○ *Adressen* [Addresses]; pp. 72–75:

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

9. Jahrgang, Heft 2/1997 (Juli 1997)

Luzia DIETSCHE, [Editorial]; p. 3

A short statement commenting on the slightly modified layout of the current issue, announcing a DANTE Web page containing the table of contents of the *Komödie*, and announcing her resignation as editor (due to major differences of opinion with the presidium).

○ *Hinter der Bühne : Vereinsinternes*
[Backstage : Club matters]; pp. 4–7:

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

A short comment on club matters by the president of DANTE: The Update CD-ROM of CTAN is distributed with this issue. Member number and invoice number together are used to identify members. And TEX Live 2 is available from DANTE.

Marion NEUBAUER, Software bei DANTE e.V. (Stand: Juli 1997) [Software from DANTE e.V. (as of July 1997)]; pp. 5–7

A short description of the software packages distributed by DANTE, updating the data given in the previous issue.

○ *Von fremden Bühnen* [On other stages]; p. 8:

Luzia DIETSCHE, Drucken wie zu Omas Zeiten
[Printing as in grandmother's time]; p. 8

A former big (East German) printing shop in Leipzig has been converted to a (hands-on) museum of the art of printing.

○ *Bretter, die die Welt bedeuten*
[The stage is the world]; pp. 9–36:

Gerd NEUGEBAUER, Überflüssiges? – Klammern um Makroargumente [Unnecessary?—Braces around macro arguments]; pp. 9–13

This article discusses in some detail when it is possible to omit the braces around arguments (as required by strict LATEX syntax), and gives an application (a special user-defined environment).

Klaus BARTHELMANN, Zähler nicht zurücksetzen
[Avoiding the resetting of counters]; pp. 13–19

Using footnotes in `report` class as an example, this article discusses how resetting of LATEX counters can be avoided.

Walter SCHMIDT, Absätze – einmal anders
[Paragraphs—done differently]; pp. 19–22

The author shows how non-zero `\parfillskip` can be used to indicate paragraphs when both `\parskip` and `\parindent` are set to zero.

Holm Sieber, Russischer Textsatz mit LaTeX [Russian typesetting with LaTeX]; pp. 22–31

The author describes the installation and use (with Russian) of the multilanguage package `babel` under teTeX.

Ernst Molitor, Aufbereitung der Ergebnisse von Literatur-Recherchen für LaTeX und BibTeX [Preparing the results of bibliographic researches for LaTeX and BibTeX]; pp. 31–36

The author describes how he has developed a tool which transforms the output of database searches (provided by library tools) into BibTeX format for use with LaTeX. The programs are written using `flex` and C++, and are available, but will have to be adapted to other record formats.

○ *Aus dem Fundus : Nützliches aus CTAN und anderen Quellen* [From the properties room: Useful tools from CTAN and other sources]; pp. 37–39:

Rolf Niepraschk, Das PSfrag-Paket [The PSfrag package]; pp. 37–39

A short description of the LaTeX package `PSfrag`, which allows the substitution of text elements in `eps` graphics.

○ *TeX-Beiprogramm* [TeX co-features]; p. 40:

LaTeX-Rätsel [LaTeX puzzle]; p. 40

A small crossword puzzle based on LaTeX terms.

○ *Leserbriefe* [Letters]; pp. 41–43:

Herbert F. Decker, Protokoll der 16. Mitgliederversammlung, Heft 1/97 [Protocol of the 16th assembly of members, issue 1/97]; pp. 41–42

This letter comments on the decreasing membership numbers mentioned in the protocol. The author's advice: Go popular! Make it easier for non-computer scientists, like teachers!

Peter Willadt, Mikrotypographie-Regeln, Teil II, Heft 1/97 [Microtypographic rules, part II, issue 1/97]; p. 43

This letter comments on two points of Marion Neubauer's article: The official standard (DIN) of the date format for office use is 1997-05-28, and according to the new German orthography, three consecutive letters will always be retained. Marion Neubauer responds.

○ *Rezensionen* [Reviews]; pp. 44–45:

Luzia Dietsche, *Lesetypographie* [Typography for reading]; pp. 44–45

This is the review of the book *Lesetypographie* by Hans Peter Willberg and Friedrich Forssmann) (Verlag Hermann Schmidt, Mainz 1997) which was given to the honorary members of DANTE. It is a handbook for the practitioner, which can be consulted in small pieces as needed. It is not only a valuable reference, but also a beautifully designed book. It is a good value, but unfortunately not cheap (DM 168,–).

○ *Spielplan* [Repertory]; pp. 46–50:

The international and national calendar, including the preliminary program of *TUG'97 — TeX Comes Home* in San Francisco (July 27–31, 1997).

○ *Adressen* [Addresses]; pp. 52–55:

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

○ *Distributed with this issue*:

CTAN Update, Stand Mai 1997, DANTE e.V. [CTAN Update, as of May 1997]

9. Jahrgang, Heft 3/1997 (Oktober 1997)

Bernd Raichle, [Editorial]; pp. 3–4

The new (interim) editor comments on the resignations of members of the *Komödie* team, differences of opinion on the responsibilities between the redaction and the presidium. This issue is delayed only because they had to wait for the welcome message of the president.

○ *Hinter der Bühne : Vereinsinternes* [Backstage : Club matters]; pp. 5–10:

Joachim Lammarsch, Grußwort [Welcome message]; pp. 5–8

This is the traditional comment on club matters by the president of DANTE. Besides the usual organizational matters (such as the installation of a closed mailing list `DANTE-EV`), the main issue is the resignation of Luzia Dietsche from her position as responsible editor of the *Komödie*. The reason for this unexpected step were major differences of opinion on editorial policy, in particular, concerning the editing of articles. Moreover, the president has ordered ("in the interest of DANTE") the placement of an article (by Marion Neubauer), and the inclusion of software lists.

Friedhelm Sowa, Kassenbericht 1995 [Treasurer's report for 1995]; pp. 9–10

○ *TEX-Beiprogramm* [TEX co-features]; pp. 11–25:

Walter Schmidt, TEX-kompatible Rechtschreibprüfung für den Editor EPM unter OS/2 Warp [TEX compatible spelling checker for the editor EPM under OS/2 Warp]; pp. 11–13

How to install (for the the editor of OS/2) a spelling checker for use in a TEX environment.

Martin Gercke, TEXWord für Windows [TEXWord for Windows]; pp. 13–15

Despite TEX's high quality, for the average computer user it has some decisive disadvantages (in comparison with Word and similar text processors). Under the motto 'Nightmare or vision?', the author describes his idea of a program TEXWord that would unite the advantages of both approaches.

Andreas Schlechte, Die neue Rächtschraibrehform [The new orthography (spelled with three mistakes!)]; pp. 15–22

A summary of the new (and much disputed) German orthography (transition period August 1, 1998, to July 31, 2005), the adaptions necessary for TEX users, and some personal (critical) comments on the reform. "When will there be the reform of the reform?"

Bernd Raichle, *Das TEXikon*, Heft 1/97: Auflösung [The TEXikon, issue 1/97: solution]; pp. 23–24

This is the explanation of the code presented in the review of the *TEXikon*.

LATEX-Rätsel, Heft 2/97: Auflösung [LATEX puzzle, issue 2/97: solution]; p. 25

The solution of the crossword puzzle in the last issue.

○ *Aus dem Fundus : Nützliches aus CTAN und anderen Quellen* [From the properties room: Useful tools from CTAN and other sources]; pp. 26–36:

Gerd Neugebauer, Vor Gebrauch schütteln [Shake before use]; pp. 26–36

This article discusses outline variants of fonts — how they can be generated with METAFONT, and how they can (and how they should) be used.

○ *Leserbriefe* [Letters]; pp. 37–46:

Henning Böke, Absätze – einmal anders, Heft 2/97 [Paragraphs—done differently, issue 2/97]; pp. 37–40

Commenting on the article cited, the author, a typographer, argues that it is best to mark paragraphs by `\parindent`. The main reason: "Not the end, the beginning of a new thought should be indicated." Moreover, he explains how paragraph indentation has developed from space reserved for the paragraph sign, ¶.

Walter Schmidt, Antwort auf den Leserbrief von Henning Böke [Response to the letter from Henning Böke]; pp. 40–41

The author of the article on unusual paragraph forms responds to the previous letter: He did not discuss the pros and cons of various paragraph shapes. He only wanted to show how an alternative effect can be obtained. However, he knows examples where the lack of paragraph indentation looks fine.

Andreas Hirsch, Zum Leserbrief von Herbert F. Decker, Heft 2/97 [On the letter by Herbert F. Decker, issue 2/97]; p. 41

This letter confirms the views of the letter cited. The problem posed by using TEX at school is not with the program itself, but with its installation: Pupils can work with TEX, but they have problems when they want to install it properly at home.

Horst Gierhardt, Zum Leserbrief von Herbert F. Decker, Heft 2/97 [On the letter by Herbert F. Decker, issue 2/97]; pp. 42–44

The author (a teacher at the German school in Istanbul) also stresses the need for tools which allow easy installation and maintainance of TEX systems (unlike the trouble caused by frequent updates to LATEX). It is mainly installation problems (in spite of Eberhard Mattes' emTEX) which prevent the use of TEX at school, or make it difficult.

Rolf Niepraschk, Zähler nicht zurücksetzen, Heft 2/97 [Avoiding the resetting of counters, issue 2/97]; p. 44

A pointer is given to two LATEX packages (by Donald Arseneau and by David Carlisle) designed to remove a counter from the list of resets. (This remark supplements the article cited.)

Wolfgang Schöbel, Ich hätt' da mal 'ne Frage … [Well, I happen to have a question …]; pp. 45–46

Another letter addressing the problems of unsophisticated LATEX users. Why should one upgrade to LATEX2$_\varepsilon$? It makes things more difficult. And the documentation of LATEX packages is not helpful enough. "Much too complicated!" — that is the most common argument used against LATEX.

○ *Von der Theaterkasse* [From the theatre box office]; pp. 48–50:

Marion Neubauer, Ältere Ausgaben von TEX-Büchern bei DANTE e.V. [Old editions of TEX books from DANTE e.V.]; pp. 48–50

A list of sale books offered by DANTE (mostly non-current editions).

○ *Spielplan* [Repertory]; pp. 52–60:

The international and national calendar, including the announcements of conferences in St. Malo (EuroTEX'98, March 29–31, and WEPT'98, March 29–April 3, 1998), and TUG'98, August 17–21, 1998, in Torun (Poland).

Announcment of the *Second Week on Electronic Documents*, March 29–April 3, 1998, in St. Malo (France).

Call for Papers for the European TEX conference, March 29–31, 1998, in St. Malo (France).

Call for Papers for the TUG TEX conference, August 17–21, 1998, in Torun (Poland),

○ *Adressen* [Addresses]; pp. 60–63:

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

9. Jahrgang, Heft 4/1997 (Februar 1998)

Gerd Neugebauer, [Editorial]; pp. 3–4

The (new) editor comments on the current situation: Because of major differences with the presidium of DANTE, many members of the redaction team have resigned. While the presidium considers the “Komödie” as newsletter of DANTE, the members of the redaction want to make it a “high quality journal”. As a consequence, several authors have withdrawn their articles. Moreover, due (only) to decisions of the president, the current issue could not be published on schedule.

○ *Hinter der Bühne : Vereinsinternes*
[Backstage : Club matters]; pp. 5–28:

Joachim Lammarsch, Grußwort
[Welcome message]; pp. 5–12

The president of DANTE starts his traditional comments on club matters by explaining the delayed publication of this issue: It was caused by a delay in the production of the CD-ROM set (delivered with this issue) which had been postponed until the LATEX update was available. After various organizational matters (software distribution, membership fees, etc.) he comments on the current situation of DANTE (resignation of members, open letter). “It is always a minority which causes troubles.”

Joachim Lammarsch, Fonds zur Unterstützung von Mitgliedern [Funds for support of members]; p. 12

DANTE is able to support members who are not able to pay their membership fees.

Marion Neubauer, Belichtungsservice
[Service for photo typesetting]; pp. 12–14

Members can take advantage of a photo typesetting service organized by DANTE.

Thomas Hafner, Protokoll der Sitzung AK Weiterentwicklung [Protocol of the meeting of the WG development]; pp. 15–16

A meeting discussing a reorganization of various services offered by DANTE.

Klaus Braune, Protokoll der Sitzung AK Software-Distribution [Protocol of the meeting of the WG software distribution]; pp. 17–18

A meeting discussing the various forms of software distribution offered by DANTE.

Uwe Münch, Protokoll der Sitzung AK Öffentlichkeitsarbeit [Protocol of the meeting of the WG public relations]; pp. 18–24

A meeting discussing various aspects of promoting TEX and DANTE (merchandising, WWW, mailing lists, etc.). The main topic, however, is the purpose of the *Komödie* and the rights and duties of the redaction team.

Bericht des technischen Beirats
[Report of the technical council]; pp. 24–28

A collection of short reports (by the appointed coordinators) on the various hardware platforms and other special topics. In particular, Jörg Knappen reports on the ec-fonts (including the tc(*text companion*)-fonts).

○ *Bretter, die die Welt bedeuten*
[The stage is the world]; pp. 29–39:

Peter Willadt, Das verflixte achte Bit
[This damned eighth bit]; pp. 29–39

The author describes how 8-bit input and 8-bit fonts change ‘daily work’ with TEX and LATEX.

○ *TEX-Beiprogramm* [TEX co-features]; pp. 40–42:

Matthias Eckermann, TEXnische Qualen
[TEXnical pains]; pp. 40–42

On a frustrating day, the author has a nightmare concerning future costs of TEX.

Markus Kohm, Jung gewohnt, alt getan
[Trained in youth, done when old]; p. 42

A short poem: Teach children Linux and LATEX!

○ *Rezensionen* [Reviews]; pp. 43–47:

Christa Post, *Erstes Arbeiten mit TEX*, von Arnulf Liebig [“Starting to work with TEX”, by Arnulf Liebig]; pp. 43–44

A review of the book (Prentice-Hall, 1996) cited in the title. The bottom line: In spite of a few flaws, it is a good introduction and a valuable reference (with an emphasis on mathematics).

Ulrik Vieth, *Schnell ans Ziel mit LaTeX2ε*, von Jörg Knappen ["Quick success with LaTeX2ε", by Jörg Knappen]; pp. 44–47

A review of the book (Oldenbourg München-Wien, 1997) cited in the title. According to the review, it is a "respactable" introduction to LaTeX which offers some unusual features, has positive and negative aspects, and sometimes is rather sloppy.

○ *Von fremden Bühnen*
[On other stages]; pp. 48–49:

Martin Schröder, TeX Merchandising — An Announcement (in English); pp. 48–49

The proposal of a project aiming at the production of TeX merchandising products like mugs, bags, T-shirts, etc.

○ *Leserbriefe* [Letters]; pp. 50–69:

Peter Schmitt, "Dezimalkomma beim TeXsatz in deutsch", Heft 1/94 ["The decimal comma with German TeX typesetting", issue 1/94]; p. 50

A small mistake in the cited article is corrected.

Peter Schmitt, "Orale Spielereien mit TeX – Teil III (Addendum)", Heft 2/96 ["Oral games — part III (addendum)", issue 2/96]; pp. 50–53

A discussion of some points concerned with a sample macro used in the article cited.

Offener Brief an das Präsidium von DANTE e.V. [Open letter to the presidium of DANTE e.V.]; pp. 53–56

A group of members (many of them founding members) criticizes the working style of the presidium. In their opinion, 'active' members who want to work for TeX and DANTE are hindered and demotivated. In particular, the bad relations between the presidium and the redaction team of the *Komödie* are discussed.

Schriftliche Antwort auf den "Offenen Brief" [Written answer to the "Open Letter"]; pp. 56–60

According to the presidium, the opinion expressed in the open letter is only that of a small group of members. The achievements of the presidium (various projects) are stressed. There is no valid reason for complaints, and the *Komödie* is 'neither a *newsletter of a club* nor a *journal of high quality content*'.

Ernst Molitor, Brief an den Präsidenten [Letter to the president]; pp. 61–63

The author criticizes the attitude of the president (as expressed in the welcome message of issue 3). The president has to *serve*, not to *order*. The president has to respect the independence of the redaction team of the *Komödie*.

Joachim Lammarsch, Kommentar zum Offenen Brief [Comment on the Open Letter]; pp. 63–65

In his personal answer the president regrets that active members have resigned. However, all the points raised in the open letter are without a real basis.

Jan Braun, Ein etwas anderer Wetterbericht [A slightly different weather report]; pp. 66–68

The author (who has compiled a database of all articles published in the *Komödie*) describes how he sees the situation, why he has signed the open letter, and why (in reaction to attitude of the presidium) he has stopped contributing to DANTE (i.e., he is no longer an active member).

Matthias Eckermann, Rücktritt [Resignation]; pp. 68–69

The author explains the reasons for his resignation from the *Komödie* redaction team.

○ *Von der Theaterkasse*
[From the theatre box office]; pp. 70–73:

Marion Neubauer, Bücher bei DANTE e.V. [Books from DANTE e.V.]; pp. 70–73

A (partial) list of books on TeX and related topics, available from DANTE.

○ *Spielplan* [Repertory]; pp. 75–76:

The international and national calendar.

○ *Adressen* [Addresses]; pp. 77–79:

Various addresses related to DANTE, and the addresses of all persons who have contributed to this issue.

○ *Distributed with this issue*:

CTAN, Comprehensive TeX Archive Network. The whole collection of LaTeX, TeX, et al.

A three CD-ROM set (archived January 1998).

(compiled by Peter Schmitt)

Calendar

1999

Jan 29 – Apr 24 *Exhibition*: Primitive types: The sans serif alphabet from John Soane to Eric Gill, St. Bride Printing Library, London, UK. For information, visit `http://www.stbride.org/soanepr.htm`.

Apr 26 – 30 XML Europe '99, Granada, Spain. For information, visit `http://www.gca.org/conf/euro99/`.

May 1 – 3 BachoTeX '99, 7th annual meeting of the Polish TeX Users' Group (GUST), "Practical Aspects of Electronic Publishing", Bachotek, Brodnica Lake District, Poland. For information, visit `http://www.gust.org.pl/BachoTeX/`.

May 11 – 14 8th International World Wide Web Conference, Toronto, Canada. For information, visit `http://www8.org`.

May 10 *TUGboat* **20** (2), deadline for technical submissions.

May 18 – 20 GUTenberg '99, "Markup and hypertext, the future of the electronic document", l'Institut de physique nucléaire de Lyon, France. For information, visit `http://www.gutenberg.eu.org/gut/manif/gut99/`.

May 24 *TUGboat* **20** (2), deadline for reports.

May 28 UK-TUG Spring meeting: TeX for Win32 systems, London, UK. For information, contact `uktug@mail.rhbnc.ac.uk`.

Jun 7 – Aug 14 ABeCeDarium: A traveling juried exhibition of contemporary artists' alphabet books by members of the Guild of Book Workers, appearing at the Ohio University Library, Athens, Ohio. Sites and dates are listed at `http://palimpsest.stanford.edu/byorg/gbw`.

Jun 9 – 11 Society for Scholarly Publishing, 21st annual meeting, Boston, Massachusetts. For information, visit `http://www.sspnet.org`.

Jun 9 – 13 Joint International Conference of the ACH/ALLC in 1999 (Association for Computers and the Humanities, and Association for Literary and Linguistic Computing), University of Virgina, Charlottesville, Virginia, USA. For information, visit `http://www.iath.virginia.edu/ach-allc.99`.

Jun 22 – 26 Student final display, Department of Typography and Graphic Communication, University of Reading, U.K. For information, call +44 (0)118/931-6398.

Jul 5 – 7 CIDE'99: Second Colloque International sur le Document Électronique, Damascus, Syria. For information, visit `http://infodoc.unicaen.fr/cide/`.

Aug 8 – 13 SIGGRAPH 99, Los Angeles, California. For information, visit `http://www.siggraph.org/s99/`.

Aug 15 – 19 **TUG'99** — The 20th annual meeting of the TeX Users Group, "TeX Online — Untangling the Web and TeX", University of British Columbia, Vancouver, Canada. The Web page, `http://www.tug.org/tug99/`, is updated regularly.

Aug 23 *TUGboat* **20** (3), deadline for reports and news items.

Aug 30 – Sep 3 Seybold San Francisco/Publishing 99, San Francisco, California. For information, visit `http://www.seyboldseminars.com/Events`.

Sep 13 – 14 EGUTH'99: First meeting of the Spanish-speaking TeX Users Group (CervanTeX), Universidad Politécnica de Madrid, Spain. For information, visit `http://feynman.faii.etsii.upm.es/~eguth99`.

Sep 19 DANTE, 21st meeting, Heidelberg University, Germany. For information, contact `dante@dante.de`.

Status as of 15 March 1999

For additional information on TUG-sponsored events listed above, contact the TUG office (+1 503 223-9994, fax: +1 503 223-3960, e-mail: `office@tug.org`). For events sponsored by other organizations, please use the contact address provided.

Additional type-related events and news items are listed in the Sans Serif Web pages, at `http://www.quixote.com/serif/sans`.

Sep 20 – 23 EuroTeX '99, the XIth European TeX Conference, "Paperless TeX", Ruprecht-Karls University, Heidelberg, Germany. Tutorials will precede and follow the main conference. For information, visit `http://uk.tug.org/EuroTeX-99/`.

Sep UK-TUG Autumn meeting and A.G.M: TeX/LaTeX and their relationship to SGML/slash HTML/slash XML, London, UK. For information, contact `uktug@mail.rhbnc.ac.uk`.

Sep 23 – 24 H2PTM99, the 5th Conference on Hypertexts and Hypermedia: Products, Tools, Methods, Saint Denis, Paris, France. For information, visit `http://www.labart.univ-paris8.fr/~conf99/`.

Oct 7 – 10 ATypI '99, Association Typographique Internationale, Boston, Massachusetts. For information, visit `http://www.atypi.org/`.

Nov 8 *TUGboat* **20** (4), deadline for technical submissions.

Nov 22 *TUGboat* **20** (4), deadline for reports and news items.

Dec 6 – 9 XML 99, Philadelphia, Pennsylvania. For information, visit `http://www.gca.org/conf/conf1996.htm`.

2000

DDEP: Digital Documents and Electronic Publishing (successor to EP98) and WEPT: Week on Electronic Publishing and Typography, to be held in the U.S.

Feb 7 *TUGboat* **20** (4), deadline for technical submissions.

Feb 7 – 11 Seybold Seminars Boston/ Publishing 2000, Boston, Massachusetts. For information, visit `http://www.seyboldseminars.com/Events`.

Feb 21 *TUGboat* **20** (4), deadline for reports and news items.

Feb-Mar DANTE 2000 and 22nd meeting, Technische Universität Clausthal-Zellerfeld, Germany. For information, contact `dante@dante.de`.

Jun 16 – 18 TypeCon 2000, Westborough, Massachusetts. For information, visit `http://tjup.truman.edu/sota/`.

Jun 22 – 24 TypoMedia 2000, "Future of Communication", Mainz, Germany. Linotype's design conference; for information, visit `http://www.typomedia.com`.

Jul-Aug **TUG 2000** — The 21st annual meeting of the TeX Users Group, in the UK.

Jul 23 – 28 SIGGRAPH 2000, New Orleans, Louisiana. For information, visit `http://www.siggraph.org/calendar/`.

Aug 28 – Sep 1 Seybold San Francisco/ Publishing 2000, San Francisco, California. For information, visit `http://www.seyboldseminars.com/Events`.

Permission for Electronic Publication of *TUGboat* material

Submission to *TUGboat* implies permission to publish on paper. However, we (the *TUGboat* editorial board) would also like to post the contents of *TUGboat* to the TUG Web pages, and hold open the possibility of creating a *TUGboat* CD. For this, we are now requesting explicit permission to accompany articles submitted to *TUGboat* for publication. This permission can be for either Web or CD or both of these electronic options.

TUG '99

TEX Online — Untangling the Web and TEX

THE 20TH ANNUAL MEETING OF
THE TEX USERS GROUP

August 15–19, 1999

UNIVERSITY OF BRITISH COLUMBIA • VANCOUVER • BC • CANADA

The 20th annual meeting of the TEX Users Group will take place at the University of British Columbia in the beautiful city of Vancouver, Canada, on August 15–19, 1999. Preliminary information is provided below. Additional information will be posted at the TUG '99 website: `http://www.tug.org/tug99/`.

The theme of this year's meeting is *TEX Online: Untangling the Web and TEX*. The focus will be on the issues involved in converting TEX/LATEX files to electronic formats. There will be talks, panel discussions, and workshops in a variety of inter-linked daily threads:

- TEX and Math on the Web
- Customizing Document Layout
- TEX in Publishing
- Fonts, Graphics, and New Developments in TEX and LATEX

Registration: will begin on Sunday August 15, 1999, in the afternoon, followed by a welcome reception the same evening. Registration includes: preprints, reception, coffee/tea breaks and lunch during the conference, and an evening banquet on Thursday. Registration forms will be available at the TUG '99 website and should be mailed or faxed to:

TUG '99, c/o Integre Technical Publishing
4015 Carlisle NE, Suite A, Albuquerque, NM 87107, USA
Fax: +1 505-889-8192 **E-mail:** `tug99@tug.org`

Registration	Until June 15, 1999	After June 15, 1999
TUG & LUG Members	390 USD	440 USD
Regular*	475 USD	475 USD
Single Day†	150 USD	150 USD
	Extra guest at banquet $60 USD	

*includes membership; †excludes banquet

Bursary Funds: may be available for support of students and other participants who demonstrate need. For bursary requests, send email to the TUG '99 Bursary Committee at `bursary@tug.org`.

Courses: will be offered, according to demand, *after* the conference on August 20–21, 1999 (and not pre-conference as previously announced). The topics will be: Introductory TEX ◦ Introductory LATEX ◦ Intermediate/Advanced LATEX ◦ Working with PostScript and Graphics ◦ Customizing TEX/LATEX Fonts ◦ Introduction to SGML and/or TEX and SGML. Other topics may be suggested. Write to Don DeLand at `tug99@tug.org`.

Sunday, August 15, 1999: ***Registration, Exhibits, and Reception***

Monday, August 16, 1999: ***TEX and Math on the Web***

Official Opening & Welcome

Stephen A. Fulling, *TEX and the Web in the Higher Education of the Future: Dreams and Difficulties*

Patrick D. F. Ion, *MathML: A Key to Math on the Web*

Doug Lovell, *TEXML brings TEX to Web Future*

Paul R. Topping, *Using MathType to create TEX and MathML Equations*

Workshop: Eitan Gurari and Sebastian Rahtz, LATEX to XML/MathML

Chris Rowley, *Beyond DVI… Is There Only PDF?*

D. P. Story, *AcroTEX: Acrobat and TEX Team Up*

Panel Discussion: Stephen A. Fulling (moderator), TEX and Math on the Web

Workshop: D. P. Story, Using LATEX to Create Quality Interactive PDF Documents for the WWW

Tuesday, August 17, 1999: ***Customizing Document Layout***

Jean-Luc Doumont, *Doing It My Way: A Lone TEXer in The Real World*

Peter Flynn, *The vulcan Package: A Repair Patch for LATEX*

Vendor Presentations: Presenting new features/releases; test driving your favorite package

David Carlisle, Frank Mittelbach, and Chris Rowley, *New Interfaces for LATEX Class Design: I & II*

Workshop: Michael Doob, Writing Class files: First Steps

Workshop: Anita Z. Hoover, Converting a LATEX2.09 Style to a LATEX 2_ε Class

TUG Business Meeting

Wednesday, August 18, 1999: ***TEX in Publishing***

Kaveh Bazargan, *Multi-use Documents: The Role of the Publisher*

Frederick H. Bartlett, *LATEX to XML/MathML for Web-Based publication*

Harry Payne, *Making a Book from Contributed Papers: Print and Web Versions*

Robert L. Kruse, *Managing Large Projects with PreTEX, A Preprocessor for TEX*

Art Ogawa, *Database Publishing with JAVA and TEX*

Paul A. Mailhot, *Implementing Dynamic Cross Referencing in PreTEX and PDF*

Hu Wang, *A Web-Based Meeting Abstract Submission System*

Petr Sojka, *Hyphenation on Demand*

Michael Downes, *Avant Garde Mathematical Typesetting*

Jonathan Fine, *Active TEX and the DOT Input Syntax*

Panel Discussion: Kaveh Bazargan (moderator), Publishers' views regarding cost effectiveness of using LATEX/TEX in publishing. Where are we now and where are we going? Is it more cost effective now because of online documentation?

Thursday, August 19, 1999: ***Fonts, Graphics, and New Developments in TEX and LATEX***

Jean-Luc Doumont, *Drawing Effective (and Beautiful) Graphs with TEX*

Wendy McKay and Ross Moore, *Convenient Labelling of Graphics, the WARMreader way*

Sergey Lesenko and Laurent Siebenmann, *Viewing DVI files with Acrobat Reader: A privileged role for DVIPDF*

Alan Hoenig, *Alternatives to Computer Modern Mathematics*

Vendor Presentations: On installing specific font packages (e.g., French, XY-pic) on various commercial implementations of TEX/LATEX

F. Popineau, *fpTEX: A teTEX-Based Distribution for Win32*

Jeffrey McArthur, *Managing TEX Software Development Projects*

Timothy Murphy, *JAVA 2000*

Panel Discussion: The Future of LATEX

Accommodation: will be available on campus at the Walter Gage Residence, University of British Columbia, or at local hotels. Arrangements must be made *directly* with the hotel or residence.

- **Gage Residence:** Reservation forms are available on the TUG '99 website, `http://www.tug.org/tug99/`, or send mail or fax (*before July 14, 1999*) to:

  Reservations Office (Group Code: G90815E)
  UBC Conference Centre
  5161 Student Union Blvd, Vancouver, BC, V6T 2C9, Canada
  Tel: +1 604 822-1010; **Fax:** +1 604 822-1001
  E-mail: `reservation@brock.housing.ubc.ca`
  `http://www.conferences.ubc.ca`

  Walter Gage Residence
  UBC Conference Centre

Room (Meals not included)	**Rate per Night**
Single Room with Shared Washroom (between 4-6 guests)	33 CAD
Premium Single Room with telephone, TV in common lounge, washroom shared between 4 guests	47 CAD
Single Suite with Private Washroom*	62 CAD
Double Suite (1 queen-size bed)*	95 CAD
Court Suite (2 twin beds, 1 queen-size bed*	110 CAD†
All rates, quoted in Canadian funds, are subject to 7% GST and 8% Hotel Tax.	

  [*] (includes TV, telephone, kitchenette and private washroom.)
  [†] Court suites are based on double occupancy.
  A charge of $10.00 per person per night will apply for each additional person.
  If requesting a Suite, please advise number of people

  Note: Private Washroom accommodation must be guaranteed with Visa, MasterCard, or bank draft in Canadian funds for the equivalent of one night. Complete payment information (including charges, deposit, etc.) is given on the reservation form.

- **Local Hotels:** Check the Lower Westside Vancouver listings on `http://www.vancouver-bc.com/Accommodations/Westside.html`.

◇ ◇ ◇

Register Now!

TUG '99 REGISTRATION:	**FAX:** 505-889-8192	**E-MAIL:** `tug99@tug.org`
c/o Integre Technical Publishing, 4015 Carlisle NE, Suite A, Albuquerque, NM 87107		
TUG & LUG Members	□ **390 USD (before June 15)**	□ **440 USD (after June 15)**
Regular	□ **475 USD**	
Name	Affiliation	
Address		
City	State/Province	Country
Phone	Fax	E-mail
Credit Card Number		Exp
Signature		Date

Late-Breaking News

Production Notes

Mimi Burbank

It is April, and the trees and flowers are blooming, and thoughts turn to new life — and the new **TeX Live** CD, which is included in this issue. Many thoughts fly through my feeble mind as I think about production of this issue — the first being that I am using the **TeX Live** 4 setup for production. I wanted to use each set of binaries available on the CD plus some extras, with the exception of the `win32`. I even managed to build a set of binaries on my own! The following binaries were used for production of this issue:

```
alpha-osf3.2
alphaev5-osf4.0d
hppa1.1-hpux10.10
i686-linux-gnulibc1
mips-irix6.2
mips-sgi-irix5.3
rs6000-aix4.1.4.0
rs6000-aix4.2.1.0
sparc-solaris2.5.1
sparc-solaris2.7
```

I tend to "play" (more proficient users would call this "beta testing") with the binaries until they are complete. I like to think I represent the "novice" when it comes to setting up a TeX system – and always tell the TL4 team members that, "If I can do it, anyone can!" I encourage all of you to read the **TeX Live** documentation, and become familiar with just what is on your system. There is a world of documentation available with all of the packages on the CD – all ready for your use and enjoyment.

Output. The final camera copy was prepared at SCRI on multiple workstations, using the *TeX Live* 4 setup, which is based on the *Web2c* TeX implementation version 7.3 by Karl Berry and Olaf Weber. PostScript output, using outline fonts, was produced using Radical Eye Software's dvips(k) 5.85, and printed on an HP LaserJet 4000 TN printer at 1200dpi.

This issue also marks a new *trend* in the production of *TUGboat* — we will also be transferring several PostScript files to the printer.

TUGboat Web Pages

I have been working to update and complete the *TUGboat* web pages for inclusion in the TL4 CD. During this process, I was overcome by guilt feelings — because we promised to begin this process some time in 1996 — and somewhere toward the end of March, I decided that we really needed to put up *all* of the contents files for every single issue of *TUGboat*! The contents are archived in special files which may be run to produce the Contents pages for each volume (or more information see `http://ctan.tug.org/tex-archive/usergrps/tug/tugboat/t-of-c/`). During the process, several comments were made about attempting to standardise the "naming" of the *TUGboat* directories on the TUG web server, so I also had to go in and rename every directory, and then edit each contents file, globally substituting one string for another. This took approximately three full days, and though the contents files are not as "nice" for the older issues, they at least are present and may be viewed at `http://www.tug.org/TUGboat/contents.html`. Any errors are my own; please send comments to `webmaster@tug.org` and I'll make changes as quickly as I have time to do so.

My primary reason for mentioning the above is to encourage you to look at the early contents files. I have been a member of TUG since 1986, and, having attended nine annual meetings between 1986 and 1996, I must admit to taking a nostalgic trip down "memory lane" as I opened and closed each of the contents files. For those who are longstanding members of TUG, there is quite a bit of history simply in the contents pages, and I enjoyed meeting "old friends" there. I only hope that we will not have to wait too much longer to be able to actually convert some of the earlier `.tex` files to a readable format on the web. This of course will require more volunteers who have time to convert some of the older files into a format which can be processed either using the original software or recoding files to run with the current version of (LA)TeX!

This will also require permission of every author who ever published in *TUGboat*! And not only for the TUG web pages. Discussion has begun with regard to making a CD of TUGboat articles. More information on this will appear on the web pages at `http://www.tug.org/TUGboat`, as well as future issues of *TUGboat*.

Coming In Future Issues The next issue will contain an article entitled "A short introduction to font characteristics" by Maarten Gelderman (reprinted with permission from the editors of the Dutch MAPS). Also scheduled for publication in the next issue is an article describing the latest Ω system.

◇ Mimi Burbank
SCRI, Florida State University,
Tallahassee, FL 32306 – 4130
mimi@scri.fsu.edu

TUG Business

Report: TUG 1999 Election

The number of candidates nominated for the open offices in the 1999 TUG election fell short of the number which would require a ballot.

The office of President was open, as were up to 12 positions on the Board of Directors. There was one candidate for President, Mimi Jett, and 11 candidates for the Board: Barbara Beeton, Karl Berry, Kaja Christiansen, Donald DeLand, Susan DeMeritt, Stephanie Hogue, Judy Johnson, Ross Moore, Cheryl Ponchin, Kristoffer H. Rose, and Philip Taylor.

According to the TUG Election Procedures, when the number of candidates is fewer than the number of open positions, all candidates who have met the qualifications are declared elected by acclamation. The term of the President expires as of the annual meeting in 2001; the terms of Board members in this class expire at the meeting in 2003. Since the Board was not at full strength before the election, the President has exercised her prerogative and appointed those candidates who were not already on the Board to assume office immediately. Please welcome Stephanie Hogue and Cheryl Ponchin. A current list of members of the TUG Board of Directors can be found inside the front cover.

Since no ballots were mailed, TEX Users Group members have not had the opportunity to read the biographies and personal statements of the candidates. Without this information it is difficult to know each candidate's particular interests, and their vision for the future of TEX Users Group. The information which would have accompanied the ballot follows this report, to introduce these individuals to the membership.

Christina Thiele
Arthur Ogawa
For the Elections Committee

Members of the TUG Board with Terms through 2003

Barbara Beeton

American Mathematical Society
P. O. Box 6248
Providence, RI 02940 USA
Phone: +1 401 455-4014
Internet: `bnb@ams.org`

Biography:

TUG: charter member of TUG; charter member of TUG Board of Directors; *TUGboat* production staff since 1980, Editor since 1983; committees: publications, bylaws, elections; chair, Technical Working Group on Extended Math Font Encoding; liaison from Board to Knuth Scholarship Committee 1991–1992

Employed by American Mathematical Society: Staff Specialist for Composition Systems; involved with typesetting of mathematical texts since 1973; assisted in initial installation of TEX at AMS in 1979; implemented the first AMS document styles; created the map and ligature structure for AMS cyrillic fonts

Standards organizations: active 1986–1997 in: ANSI X3V1 (Text processing: Office & publishing systems), ISO/IEC JTC1/SC18/WG8 (Document description and processing languages); developing the standard acroISO/IEC 9541:1991 Information technology — Font information interchange

AFII (Association for Font Information Interchange): Board of Directors, Secretary 1988–1996

STIX representative to the Unicode Technical Committee for adoption of additional math symbols

Personal statement:

TUG has changed over the years, with its transition from an appointed to an elected Board. Those charged with shaping its future direction have tried to do so in a way that encourages participation by all members, not just a few. Similarly, the typographic landscape has changed as well, and though the object that is our focus — TEX — is still a tool of undeniable utility, it is just part of a growing pool of text processing software, some of it borrowing from the features that first attracted us to TEX. I maintain my commitment to Don Knuth's original goals for this tool: high typographic quality and portability. Within this framework, my goal is to continue working for unconstrained communication among TeX users, to encourage exploration of techniques consistent with the typographic excellence we have come to expect, and to act as a historian of the TeX community when that is appropriate.

Karl Berry

135 Center Hill Road
Plymouth, MA 02360
Internet: `kb@cs.umb.edu`

Biography:

First contact in 1982. Subsequently, many installations at many organizations (not to mention many readings of the *TeXbook* and *Metafontbook*). Co-author of *TeX for the Impatient*, one of the first comprehensive non-Knuthian books on TeX. I took over maintenance of the Unix port of TeX (i.e., Web2c) in 1990 from Tim Morgan (and have since happily given it to another volunteer). Along with Web2c, I developed kpathsea, a freely redistributable library for path searching and variants of three DVI drivers that use it; Eplain, a macro package that extends plain TeX(and an unpublished alternative to LaTeX based on it); `modes.mf`, a collection of Metafont modes and adaptations; a list of short fontnames for portable use within TeX across platforms; and adapted the Adobe Lucida Math fonts for TeX. I am also the maintainer and primary developer for GNU Texinfo, a TeX-based documentation format. Besides such programming tasks, I've also produced the usual books, articles, collections, and ephemera, studied typeface design, and co-written several articles on reading research and mathematical analysis of type. For TUG, I serve as the system coordinator for Unix, and participate on several committees (notably CTAN, DVI standards, and TeX directory structure), as well as acting as principal system administrator of the `tug.org` and `ctan.org` machines and domains.

Personal statement:

For TeX to grow, and perhaps even to survive, I believe there must be substantive development of the basic TeX program, and that TUG should support such. I am particularly interested in furthering the cause of TeX as a public program and competitive alternative to commercial typesetting programs.

Kaja P. Christiansen

Department of Computer Science
University of Aarhus
DK-8000 Århus C, Denmark
Internet: `kaja@daimi.au.dk`

Biography:

I was born in Warsaw, Poland. After obtaining an MSc in Mathematics at the University of Warsaw, I eventually moved to Denmark. I came to love my new country, where I have now lived and worked for more than 20 years.

My job at the Department of Computer Science of the University of Aarhus involves system administration, system and software support for our SUNs and responsibility for all aspects of a well-functioning TeX system on our Unix and Macintosh platforms: maintainance, local styles, in-house classes and (very) frequent user support, both at our department and others. The department has about 550 students, 80 employees, a large number of active research groups, close ties to the National Centre for IT Research (CIT), and it hosts the BRICS Research Centre and International PhD School.

Personal statement:

The first time I heard about TeX was in 1979. On leave in Palo Alto, I wanted to take some courses at Stanford and my top priority was lectures by Prof. Donald Knuth. That's impossible, I was told, Prof. Knuth was on leave due to work on *a text processing project...* Back home, it didn't take long before we had installed a runnable system and thus introduced an early version of TeX in Denmark. Times were different: we were on friendly terms with TANGLE and WEAVE, and local modifications (Danish!) were managed with our own style files. Since then, TeX (and later LaTeX) has been used by our faculty, students and staff for research publications and all sorts of documents.

TeX is more than a task to me — it is also a hobby. Much has changed over the years. There are impressive developments within package libraries, font area and multilingual support — just to name a few. New tools are designed for a portable and standarized system. In an ever changing and dynamic field, TeX remains to be an irreplacible tool, used and loved by a worldwide TeX community. As a member of the board, I am committed to help promoting TeX related projects. My special interests are projects of immediate value to TeX users: TeXLive, *TUGboat* and TUG's WEB site, and I would like to continue working on them.

Don DeLand

Integre Technical Publishing Co., Inc.
4015 Carlisle NE, Suite A
Albuquerque, NM 87107
Internet: `don@integretechpub.com`

Biography:

In 1991 I founded Integre Technical Publishing, a full-service composition house that specializes in producing STM (science, technical, and medical) books and journals. The core of our business has been the production of books and journals using TEX and LATEX, and we are currently developing full-text SGML and online projects using a variety of TeX-related tools such as IBM's techexplorer, Scientific WorkPlace, and Advent 3b2.

I have served as treasurer and board member of the TEX Users Group since the summer of '97. As treasurer I have been working with the office staff to put TUG's financial records in order, create a budget, and generate a realistic picture of TUG's financial situation. I have also been involved in organizing the TEXNE and TUG'99 conferences.

Personal statement:

There are three goals I would like to see TUG accomplish in the coming years:

(1) Make TUG more responsive to the needs of its members. TUG needs to publish articles and present conferences that are relevant to the average TEX user; we need to revive the teaching of courses; and we need to ensure the continued delivery of the CTAN and TEX Live CDs.

(2) Increase TUG membership and revenues. New members are necessary to the vitality of TUG. TUG can attract and retain members by increasing responsiveness, distributing *TUGboat* in a timely manner, and increasing TUG's visibility particularly within math/science societies and at colleges and universities. With a larger membership TUG could better afford to sponsor specialized conferences and subsidize developmental efforts that members have requested.

(3) Continue improving the organization of TUG and the TUG office. The TUG office has made herculean efforts in the past year toward better organization and improved response time to inquiries, but a great deal of work remains. In addition, TUG's archives for membership, finances, and past board proceedings are not well organized, and there currently exist no organizational guidelines for the board. The result has been inadequate transition between successive boards.

Susan DeMeritt

Center for Communications Research
Institute for Defense Analysis
4320 Westerra Court
San Diego, CA 92121
Internet: `sue@ccrwest.org`

Biography:

I have been employed by the Center for Communications Research – La Jolla (CCR–LJ) since its inception in 1989, as the Senior Technical Publications Specialist. My responsibilities include coordinating the production of technical working papers and organizing conferences. I manage the inputting, editing, and publication preparation of working papers for 30 full-time (in-house) mathematicians, plus 20–30 more who participate in our summer program. Our authors use various forms of TEX as well as LATEX!

I also assist with organizing conferences and meetings for groups of people numbering anywhere from 10–75.

Personal statement:

As a member of the Board of Directors of the TEX Users Group, I enjoy meeting TEX users from around the world and I enjoy the challenge of being involved in the workings of the group.

I have particularly enjoyed working on the annual (and semi-annual) conferences. I have been involved, at increasing levels of responsibility, with the conferences in San Francisco and New York City, and I am currently participating in planning the 1999 annual conference in Vancouver. Having worked on these conferences, as well as others for CCR-LJ, I am very motivated to keep improving the quality of the annual meeting so that everyone from the developer to the end user will benefit. I also want to work on assuring that the cost of the annual meeting allows for as many people to attend as possible.

Stephanie Hogue

The TypeWright
801 Highland Road
Lansdale, PA 19446 U.S.A.
Internet: `shogue@typewright.com`

Biography:

For over 14 years, I worked for the Finance Department of the Wharton School at the University of Pennsylvania. In the '80s, we used proprietary word-processing and desktop-publishing equipment to produce research papers, data tables, and graphs for 30–40 professors. I managed this office and was also responsible for the department's LAN. When Word Perfect announced its first equation editor, the Finance Dept. formed a committee to consider the feasibility of migrating from the word-processing system to PC's. Appointed to this committee by the Department's Chairman, I was not impressed with the equation editor, which one of the assistant professors characterized as "watered-down TEX." He proposed adopting LATEX and showed me a file, cautioning me that "It's not as bad as it looks at first!" Once I tore my eyes away from that preamble, I agreed that LATEX seemed to have much more potential than Word Perfect.

About four weeks after starting to use LATEX, I received the first request for a style change! "Please figure out how to change the footnote mark for `\thanks` from a number to an asterisk when the titlepage option is used in an article." Thinking that all these commands are defined somewhere, I started rummaging in the style files. Comparing the definitions of `\maketitle` with and without the titlepage option led to the necessary code, and that was my first hack.

My job at Wharton evolved into providing TEX support — macro-writing, assisting users, installing and upgrading — as well as processing research and class materials. In 1996, I left Wharton and started freelancing. I now do business as "The TypeWright"; most of my clients are academicians in the fields of finance and econometrics. Last year, I had the chance to work with some terrific volunteers organizing the TEXNortheast conference, and I am currently Co-Chairman of the Program Committee for the 1999 meeting in Vancouver.

Personal statement:

While TEX is unsurpassed at mathematical typesetting, and the latest developments are exciting, and TEX on the Web offers new possibilities, the sheer amount of information is overwhelming. I think the challenge facing TUG is to help users access and understand what is available.

The enthusiastic response to TEXNortheast indicates that people need programs of this kind. I am interested in helping to design such conferences and in revitalizing TUG's training programs.

Judy Johnson

600 SW Spring Lane
Portland, Oregon 97225
Internet: `jannejohnson@yahoo.com`

Biography:

I have spent the last 8 years working directly with TEX and LATEX in commercial book publishing. My years of experience as a board member has turned my focus to the operations of the TUG Office.

Personal statement:

I wish to continue on in that direction, as I feel that our members and the service that we provide to them is our most important goal. We have established a cohesive office staff, corrected and updated our member data base, and are currently working towards making training available again. We are eager and excited about the future of the TUG Office!

Ross Moore

Mathematics Department
Macquarie University
Sydney, Australia
Internet: `ross@mpce.mq.edu.au`

Biography:

After obtaining a first degree from the University of Melbourne, Ross studied Mathematical Physics at the University of Oxford, emerging with a doctorate in 1981. Then followed 5 years at the Australian National University, before moving to Sydney in 1986 to work at Macquarie

University where he now teaches Mathematics, with an emphasis on Geometry and Computing.

Ross's current work is mainly about developing software useful for the presentation, display and publication needs of the scientific community, and mathematicians in particular. (For those in the know, this means TEX and related products.) Notable in this area is his contribution to XY-pic, being a primary author along with Kristoffer Rose. His work on LATEX2HTML during the past few years has made this into one of the most versatile and easy-to-use interpreters for LATEX, adding an important tool for the creation of Web pages, for academics and scientists, starting from LATEX source.

Ross joined the TUG board of directors at the 1997 meeting, in San Francisco and attended the TEX-NorthEast meeting in 1998. He frequently gives help and advice on the Textures and LATEX2HTML discussion lists.

Personal statement:

I have expertise programming in several languages: e.g. PostScript, Perl, HTML, *Mathematica* as well as TEX and LATEX, and have experience in combining these, as is needed in the production of technical documents.

As a member of the TUG board of directors, I intend to continue to offer my services and experience in these areas, and to help promote the use of TEX, LATEX and related products, as a useful tool for mathematicians, scientists, students and others.

Cheryl Ponchin

Center for Communications Research
Institute for Defense Analyses
29 Thanet Road
Princeton NJ 08540-3699
Internet: `cheryl@ccr-p.ida.org`

Biography:

My name is Cheryl Ponchin and I am employed at the Center for Communications Research–Princeton (CCR-P) as a Technical Document Specialist III. My responsibilities at IDA/CCR-P include the production of technical working papers. I do papers for many mathematicians, plus many others who participate in our summer program. Our authors use various forms of TEX as well as LATEX!

Personal statement:

I have previously worked on the San Francisco and New York City conferences. I am currently participating in planning the 1999 annual conference in Vancouver. I have enjoyed working on these conferences very much and I would like to continue to offer my services to the TEX community.

Kristoffer Høgsbro Rose

LIP, École Normale Supérieure de Lyon
46 Allée d'Italie
69364 Lyon 7, France
Internet: `Kristoffer.Rose@ens-lyon.fr`

Biography:

I am a researcher in computer science, more specifically in the theory of programming languages. I began working with TEX in 1984, getting the TEX82 software from Stanford and making it work with Nelson Beebe's DVI driver suite. Once I started macro programming with TEX I was hooked: the fact that one can write formulae and programs that directly result in beautifully typeset output fascinates me still. The largest TEX thing I have written, and the one that got me onto the TEX stage, is a drawing system called XY-pic. One of its most important features is that one can easily produce drawings by automatic means, something that I hope to look into more in the future. The development of standardised markup formats in the SGML/XML context makes this realistic now, I think, so I have work ahead of me.

There is more information about me (and XY-pic) on my home page, `http://www.ens-lyon.fr/~krisrose`.

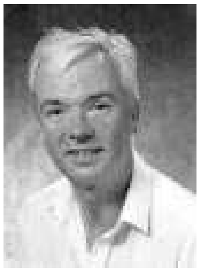
Personal statement:

I believe that the world is finally maturing to a point where documents are available in electronic form: instead of each document being an incomprehensible monolithic binary-format blob of data, it should be searchable, categorisable, etc. We in the TEX community are used to the luxuries provided by text files with well-defined markup. However, it has taken more than ten years for SGML to become accepted. A workable standard for mathematics, MathML, is now emerging, thus SGML is quickly invading the domain where TEX has been reigning for more than a decade.

This is a huge challenge for us in the TEX community, and many principles will have to be bent in the time to come. For example, should we encourage TEX engines to convert directly from SGML to PDF, completely bypassing text with \s and the venerable DVI format that we all know and love? I believe so, and hope that it will be as easy to interchange documents with advanced mathematics (and drawings!) as it is to exchange textual mail. I am certain that TEX has its place as the typographical engine producing output for a multitude of devices in the highest quality available for mathematical typesetting.

In my opinion we are ready to tackle that challenge in earnest; and the two international TEX conferences this year, TUG and EuroTEX, both have themes related to electronic publication.

Philip Taylor



The Computer Centre
RHBNC, University of London
Egham Hill
Egham, Surrey TW20 0EX
United Kingdom
Internet: P.Taylor@vax.rhbnc.ac.uk

Biography:

Philip Taylor has been a TEX devotee ever since a visit (many years ago) to British Petroleum where he saw equipment *identical* to that which he was using (a Digital VAX, with Digital LN03 printer) producing output infinitely better than anything he could manage. Having found that BP were accomplishing this using TEX, he immediately took a copy on magnetic tape, and has never looked back. Although he now spends more time on electronic publishing than on typesetting *per se*, he is still a staunch advocate of TEX, and is currently Chairman of the UK TEX Users' Group, a member of the TUG Board, Programme Committee Chairman for EuroTEX'99, and Technical Director of the $\mathcal{NTS}$ project. He believes that, above all else, TUG should remember that its *raison d'être* is TEX, and that TUG should therefore concentrate on the furtherance of TEX rather than its own existence. If elected to the Board, he undertakes to continue to press for the needs of the member to be put before the needs of the TUG office.

Position of President

Mimi Jett

IBM
T. J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
Internet: mimi@teleport.com

Biography:

Mimi Jett started her first business in the graphics and publishing industry in 1977, The Renaissance Press. Ten years later Electronic Technical Publishing (ETP) was her start-up that grew from the basement into a full-service provider of technical book composition and related services to college and professional publishers. For the past year, she has been working closely with Interactive Composition Corporation on the evolving requirements of publishing on the Internet. Beginning in April, Mimi will become the techexplorer Evangelist for Advanced Internet Publishing at IBM. Mimi and her husband Mike have four daughters ranging from 17 to 27 years of age, and two 8-month old puppies.

Personal statement:

Working for TEX Users Group over the past two years has been much more challenging, but also more rewarding, than I had imagined it could be. This is such an exciting and diverse combination of people, with so many opinions, ideals, and creativity.

It was through TUG that I met Gordon Johnson, the owner of Interactive Composition. Gordon is a very special man, who has contributed greatly to the success of many people, myself included. ICC continues to provide office space to TUG rent-free. Thank you, Gordon and Scott! It was also through TUG that I was introduced to the work IBM has done with techexplorer — utilizing LATEX or MathML code in a dynamic hypermedia display — which changed my outlook on publishing, and subsequently, my life. I am very excited now to work with IBM, a truly great company.

It is an honor to represent our organization in the world community. If elected president, I will continue to work for the benefit of all our members. Thank you for your support.

Institutional Members

American Mathematical Society,
Providence, *Rhode Island*

CNRS - IDRIS,
Orsay, *France*

College of William & Mary,
Department of Computer Science,
Williamsburg, *Virginia*

CSTUG, *Praha*, *Czech Republic*

Florida State University,
Supercomputer Computations
Research, *Tallahassee*, *Florida*

Hong Kong University of
Science and Technology,
Department of Computer Science,
Hong Kong, *China*

IBM Corporation,
T J Watson Research Center,
Yorktown, *New York*

ICC Corporation,
Portland, *Oregon*

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*

Kluwer Academic Publishers,
The Netherlands

KTH Royal Institute of
Technology, *Stockholm*, *Sweden*

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*

Max Planck Institut
für Mathematik,
Bonn, *Germany*

New York University,
Academic Computing Facility,
New York, *New York*

Princeton University,
Department of Mathematics,
Princeton, *New Jersey*

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

Stockholm University,
Department of Mathematics,
Stockholm, *Sweden*

University of Canterbury,
Computer Services Centre,
Christchurch, *New Zealand*

University College,
Computer Centre,
Cork, *Ireland*

University of Delaware,
Computing and Network Services,
Newark, *Delaware*

Universität Koblenz–Landau,
Fachbereich Informatik,
Koblenz, *Germany*

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

University of Texas at Austin,
Austin, *Texas*

Università degli Studi di Trieste,
Trieste, *Italy*

Uppsala University,
Computing Science Department,
Uppsala, *Sweden*

Vrije Universiteit,
Amsterdam, *The Netherlands*

TeX Consulting & Production Services

Information about these services can be obtained from:

TeX Users Group
1466 NW Naito Parkway, Suite 3141
Portland, OR 97209-2820, U.S.A.
Phone: +1 503 223-9994

Fax: +1 503 223-3960
Email: `office@tug.org`
URL: `http://www.tug.org/consultants.html`

North America

Hargreaves, Kathryn
135 Center Hill Road,
Plymouth, MA 02360-1364;
(508) 224-2367; `letters@cs.umb.edu`

I write in TeX, LaTeX, METAFONT, MetaPost, PostScript, HTML, Perl, Awk, C, C++, Visual C++, Java, JavaScript, and do CGI scripting. I take special care with mathematics. I also copyedit, proofread, write documentation, do spiral binding, scan images, 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 some psychophysics research papers. I have an MFA in Painting/Sculpture/Graphic Arts and an MSc in Computer Science. Among numerous other things, I'm currently doing some digital type and human vision research, and am a webmaster at the Department of Engineering and Applied Sciences, Harvard University. For more information, see: `http://www.cs.umb.edu/ kathryn`.

Loew, Elizabeth
President, TeXniques, Inc.,
675 Massachusetts Avenue, 6th Floor,
Cambridge, MA 02139;
(617) 876-2333; Fax: (781) 344-8158
Email: `loew@texniques.com`

Complete book and journal production in the areas of mathematics, physics, engineering, and biology. Services include copyediting, layout, art sizing, preparation of electronic figures; we keyboard from raw manuscript or tweak TeX files.

Ogawa, Arthur
40453 Cherokee Oaks Drive,
Three Rivers, CA 93271-9743;
(209) 561-4585
Email: `Ogawa@teleport.com`

Bookbuilding services, including design, copyedit, art, and composition; color is my speciality. Custom TeX macros and LaTeX 2_ε document classes and packages. Instruction, support, and consultation for workgroups and authors. Application development in LaTeX, TeX, SGML, PostScript, Java, and ßC++. Database and corporate publishing. Extensive references.

Outside North America

DocuTeXing: TeX Typesetting Facility
43 Ibn Kotaiba Street,
Nasr City, Cairo 11471, Egypt
+20 2 4034178; Fax: +20 2 4034178
Email: `main-office@DocuTeXing.com`

DocuTeXing provides high-quality TeX and LaTeX typesetting services to authors, editors, and publishers. Our services extend from simple typesetting and technical illustrations to full production of electronic journals. For more information, samples, and references, please visit our web site: `http://www.DocuTeXing.com` or contact us by e-mail.