TUGboat

Volume 17, Number 1 / March 1996

TeX Users Group

Memberships and Subscriptions

TUGboat (ISSN 0896-3207) is published quarterly by the TeX Users Group, Flood Building, 870 Market Street, #801; San Francisco, CA 94102, U.S.A.

1996 dues for individual members are as follows:

- Ordinary members: $55
- Students: $35

Membership in the TeX Users Group is for the calendar year, and includes all issues of *TUGboat* for the year in which membership begins or is renewed. Individual membership is open only to named individuals, and carries with it such rights and responsibilities as voting in the annual election. A membership form is provided on page 86.

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

Second-class postage paid at San Francisco, CA, and additional mailing offices. Postmaster: Send address changes to *TUGboat*, TeX Users Group, 1850 Union Street, #1637, San Francisco, CA 94123, U.S.A.

Institutional Membership

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

Printed in U.S.A.

Board of Directors

Addresses

All correspondence,
payments, parcels,
etc.
TeX Users Group
1850 Union Street, #1637
San Francisco,
CA 94123 USA

If you are visiting:
TeX Users Group
Flood Building
870 Market Street, #801
San Francisco,
CA 94102, USA

Telephone

+1 415 982-8449

Fax

+1 415 982-8559

Electronic Mail

(Internet)

General correspondence:
`TUG@tug.org`

Submissions to *TUGboat*:
`TUGboat@AMS.org`

TeX is a trademark of the American Mathematical Society.

I submit that great typography is a product of hard work, knowledge and diligence, and developments in technology only alter what it is that one must work hard at, have knowledge of and be diligent about.

John Hudson, Type Director, Tiro TypeWorks, Vancouver, BC
Message to `comp.text.tex`
(November 23, 1995)

COMMUNICATIONS OF THE TEX USERS GROUP
Editor Barbara Beeton

Volume 17, Number 1 • March 1996
San Francisco • California • U.S.A.

Soliciting Bids for TUG '97

TUG is soliciting bids for volunteers to host the annual meeting in 1997. All bids will be considered and the final choice will be based on price and availibility of services common to TUG annual meetings.

A checklist is available for completion by candidates, which should then be returned to the TUG Executive Committee. Along with this checklist, TUG can provide organizers a set of useful documentation based on accumulated experiences of previous TUG meeting organizers. The checklist may be retrieved from the CTAN archives in ASCII format: `tex-archive/usergrps/cpcform1`.

If you are interested in hosting an annual meeting at your site, please contact TUG by email at `tugex-l@irlearn.ucd.ie` or by post to the TUG office at:

TeX Users Group
ATTN: Conference Planning
1850 Union Street, #1637
San Francisco, CA 94123 USA

From the President

Michel Goossens

After those long and cold winter months we all long for Spring, with its flowers, sunshine, and warmth that make us dream about our next summer holidays. Since last September the members of the *TUGboat* Production Team have been working hard to produce five issues of *TUGboat* in as many months. I think we have shown that TUG is still alive and well, and can still deliver the goods it promises. Therefore I hope, dear reader, that we have regained your complete confidence, and that we shall be able to count on your continued support, since TUG exists *through and for its members.*

This is your first issue of *TUGboat* for 1996 and, as announced in the December issue last year, the Proceedings of TUG '96 will come next, followed by another regular issue in September. In December we plan to have a special thematic issue, exploring examples of how TEX is used around the world in non-mathematical contexts to generate real "master works of the typesetting Art". Barbara Beeton, in her comments, gives a few more details. We invite all those interested to submit an article for any of these issues, because I am convinced that numerous TEX and METAFONT pearls are still hidden in many parts of Planet Earth.

As each year, one of the great TEX events of 1996 will without doubt be the annual TUG Conference. As announced several times before, and repeated one last time on page 76, TUG '96, our 17th Annual Conference, will take place from Sunday July 28th to Friday August 2nd in Dubna, Russia. Dubna is a small but famous town a mere 70 miles north of Moscow, in the middle of picturesque pine woods, on the banks of the Volga, and the home of an international scientific institute and a university. Those attending TUG '96 will be able to meet many TEX gurus and celebrities, or solve that stubborn TEX-problem that has been with them for so long, or talk face-to-face with the email correspondent only known from electronic exchanges. It will also be a unique occasion to get acquainted with a different but still profoundly Indo-European culture, its traditions, gastronomy, and the typical Slav hospitality. You will return home with a feeling of well-being, that you were amongst those who participated in the first-ever TUG Conference organized outside the English-speaking world, at the frontier where East meets West, where Europe ends and Asia starts, a real melting pot of cultures. Where "da" means "yes", and "nyet" means "no" — and all that served in the Cyrillic alphabet. I look forward to meeting all of you, TEX-enthusiasts, in Dubna in July at TUG '96. You will not regret your visit!

Let me come back one moment to the present issue of *TUGboat*, where you can find a wide variety of topics. First of all we include the transcript of Donald Knuth's question and answer session at TUG '95. It gives an in-depth insight into the way of work of the author of "The Art of Computer Programming", and the father of TEX. Let me wholeheartedly thank Christina Thiele for her painstaking editing of Don's text, based on her personal notes and a tape recordings, and Don himself, who took the time to carefully go through the proofs only a few days before he had to leave for Europe to lecture at various universities, and to receive a degree of *Doctor honoris Causa* of Masaryk University in Brno (Czech Republic). Congratulations to Don who, by his continuous efforts for writing about and teaching mathematics, and his professional and thorough approach to problem solving, is and will remain an example of true dedication to the ideals of pure science for generations to come.

As Barbara Beeton, the *TUGboat* Editor, and I already explained, we want *TUGboat* to become a journal that can be read by TEX users of all levels of TEX proficiency. Therefore, we welcome tutorials like the one on LATEX 2ε's graphics capabilities by Ken Reckdahl. We also plan to regularly include material about what is happening in the various TEX user groups worldwide. In particular, an overview of the resources which these groups have to offer can be found in the present issue.

As you will have realized by reading the editorials in past *TUGboat*s, TUG has been going through a difficult period. We hope that the worst is now behind us, so that we can concentrate on building for the future. First of all let me reassure you that TUG, in this seventeenth year of its existence, will continue its quest for finding optimal ways to support the TEX community worldwide. I remain convinced that *TUGboat* will continue to play an important role in TUG's future as *the* reference journal with TEX-related material, but I am equally sure that other, complementary communication channels will become ever more important, as indicated by the many articles on TEX, electronic documents and the Internet in recent issues of *TUGboat*. Therefore it is with great pleasure that I am able to announce that Karl Berry (the author of the well-known TEX implementation `web2c`, plus many software packages in the GNU world) has graciously put a PC running linux at TUG's disposal. It is connected to the Internet at Karl's university (the University of Massachusetts at Boston, also known by its acronym `umb`). We run a WWW and ftp service on that machine (see the article on the resources offered by the various TEX user groups on page 37). Let me take this occasion to express TUG's sincerest thanks to Karl for allowing us to use his machine, and to his University for their Internet connection. With time I am convinced that the names `www.tug.org` and `ftp.tug.org` will become part of the daily (electronic) vocabulary of every TEX user as a convenient one-stop site, where up-to-date reference information on TEX, TUG and the other User Groups can be found.

The Future of the TEX Users Group

At the end of my first message of 1996 to you, dear Reader, it is also my duty to inform you that the storm clouds we have been living under for the past year have not gone away. TUG, and especially the *TUGboat* Production Team, have done their best to deliver *TUGboat* on schedule again, hoping that this would lead to a new wave of renewals. We had hoped that we would be able to reach a number of at least 2000 members again, the minimum TUG needs to survive in its present form. The mid-March membership figures stand at around 1200 (57% North America, 35% Europe and 8% rest of the world). Unless we witness a significant increase in the renewal rate in the coming months, we will have to take drastic measures to reduce cost even further, and probably the TUG Office will have to be closed down temporarily, to allow us to concentrate on the production of *TUGboat* and seeding TEX-related activities worldwide, thereby solely relying on volunteer labor.

TUG realizes that computing changes at an ever-increasing pace and what was common practice only yesterday is already old stuff and unusable today. The Internet, CD-ROMs, multi-Gbyte hard-disks attached to a PC equipped with eight or more Mbytes of memory are now commonplace. TUG must and *will* find its place in this new Web-linked universe. We shall concentrate on areas where we can positively contribute, namely electronic publishing, by producing *TUGboat*, *the* definitive international journal on TEX-related issues, and coordinating the information flow between the various local TEX user groups worldwide, which, by exploiting the proximity factor, are, in general, better placed to serve their respective user bases. Therefore, acting as a clearing house for global TEX-related information and complementarity with the local TEX Groups will be our motto for the future. I sincerely hope that we will be able to count on all friends of TEX wherever they reside to work together to support us in this endeavor.

◇ Michel Goossens
CERN, Geneva, Switzerland
`goossens@cern.ch`

Editorial Comments

Barbara Beeton

DEK on tour

As we go to press, Don Knuth is touring parts of the Continent, stopping in various places to receive honors, give speeches and answer questions. Every time he answers questions about TEX in front of a TEXnical audience, a little more history emerges. Some of that appears later on in this issue — a transcript of Don's Q & A session at last summer's TUG meeting in Florida. And tapes and transcripts are being made during the the current journey — something we might look forward to in issues to come.

Don has also re-established himself on the Internet, but not as an e-mail recipient. An attractive and informative set of WEB pages can be viewed at `http://www-cs-faculty.stanford.edu/~knuth/`; in them you can learn about what he's doing these days, why he won't read e-mail (or faxes), and many other facts important or trivial.

One important fact is the situation with the Computer Modern fonts. Corrections were made in 1992, but many people are not yet using the "new" versions. An "Important Message", extracted from Don's WEB pages, appears on p. 29.

Don's words have also recently appeared in the April 1996 issue of *Dr. Dobb's Journal*, where Jack Woehr records "An Interview with Donald Knuth". Check it out for some history of computer languages *other than* TEX. (Thanks to Nelson Beebe and Lincoln Durst for bringing this article to my attention.)

TUGboat: the year ahead

The publication schedule will be somewhat different this year, with the Proceedings being compiled and published (as issue no. 2) *before* the annual meeting and copies handed out at registration. In past years, the authors' productions were compiled first as preprints, and only after the meeting frozen in the formal *TUGboat* format. However, the committee for this year's meeting preferred (as is traditional in some circles) to be able to distribute the formal proceedings at the meeting. Any papers not delivered for processing by June 10 will not be included, but held for a later issue.

Issue 3 will be a regular issue.

Issue 4 will be a theme issue, on "non-mathematical" TEX, TEX and the humanities, TEX and the world's languages, ... Two decidedly non-mathematical TEX users, Pierre MacKay and Christina Thiele, both former TUG presidents, have agreed to be co-editors of this special issue. They will be starting with some ideas of their own, but would welcome suggestions and proposals from readers and prospective authors. They can be reached at the addresses shown in the address list at the front of this issue.

The first issue of 1997 — volume 18! — will again be a regular issue.

Suggestions for topics that might be appropriate for future theme issues are welcome; they can be sent to me or to any member of the *TUGboat* team, listed on the back of the title page.

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

TUG'95
Questions and Answers with
Prof. Donald E. Knuth

Don Knuth with past (and present) TUG presidents [Malcolm Clark absent]. (l–r) Pierre MacKay, Bart Childs, Don Knuth, Nelson Beebe, Christina Thiele and Michel Goossens (Photo courtesy Luzia Dietsche).

Session called to order by Barbara Beeton.[1]

Barbara: I've had the pleasure of knowing Don for quite a long time. I'd like to start off with the first question ...[2] the obvious question, *other* than what the T-shirt means: How's volume 4 doing? [laughter]

DEK: Thank you very much, Barbara. You said that I'm the reason that most of you are here. I think that Barbara is just as much a part of the reason, as me or anyone, about why we're here. She's done wonderful work for us all these years.

You know, the reason I came to this meeting is that, after the 10th TUG conference, I promised I would come to the 16th because that was the most important number for a computer scientist. Sixteen is not only a power of two, it's two to a power of two to the two, and 16 is two to the two to the two to the two to the two to the minus-infinity. [laughter] So, it's about as binary a number as you can get until 65,536. Numbers are important to me. So this is a momentous meeting for the whole TEX project.

[1] This Q&A session took place at the TUG'95 Annual Meeting in St. Petersburg, Florida, on 25 July 1995 (9–10:30 a.m.) –Ch.

[2] Transcription notes: (a) ... means a bit of a pause in the speaking; (b) [...] means text missing because unclear or unintelligible.

I looked up what was I doing exactly 16 years ago today. And I found out that, of all things, I was working with Barbara Beeton, who had come to Stanford for 2 or 3 weeks for the American Math Society. They were showing me the problems they were having with making the index to *Math Reviews*. So, on July 25th, 1979, Barbara and I were trying to figure out how to do the index to *Math Reviews*. This led to more powerful facilities for leaders and things like that, because the index to *Math Reviews* has occasions where you have lots of problems because you have to run the dots in a certain way, based on how many references there are. It was very interesting because I found two bugs in TEX that day — they were numbers 414 and 415 in the history of the development of TEX. Something to do with an error message in case you get to an end-of-file in the middle of something else. Anyways, it was that kind of error. That was sixteen years ago today.

Barbara said: "Why do I have this T-shirt on?" The T-shirt says:

$$x^n + y^n = z^n \quad \ldots\ \textit{NOT!}$$

That's a mathematical formula which I could show you how to do in TEX, if you're interested ... [laughter]

I'm wearing this T-shirt because I had a thrill a month ago. It's continually exciting for me to see the uses that people are making of TEX all over the world. Very exciting. One of the most important somehow to me was last month when I went to the library and saw Andrew Wiles's solution to Fermat's Last Theorem. I think a lot of you know that it was front-page news.[3] For a while, there was some doubt whether there was maybe still a gap in his proof, and then it was fixed up. So, this is the most famous by far of all problems in mathematics. Just as people can remember where they were when they heard about Kennedy being assassinated, I know mathematicians can all remember where they were when they first heard that Fermat's Theorem was solved. The paper came out in the *Annals of Mathematics* last month;[4] it arrived in our library and I saw it sitting there, and I looked at it and it was just wonderful for me because it was in TEX and it looked gorgeous! [laughter] This to me was the ... you know, it was so ... I mean, I almost felt like I had helped to solve the Theorem myself! [more laughter] So now, I'm also very glad to find out that the people who were responsible are here this week. In the back row, we have the editor of *Annals of Mathematics*[5] and also Geraldine Pecht, who was the typesetter. So, it's a thrill to me. Let's give them a hand. [applause]

But I didn't want to talk about anything prepared, I wanted to answer questions.

So, Barbara asked the first question: "What about Volume 4 of *The Art of Computer Programming*?" Now, I usually only answer that question on special occasions. [laughter] These days I'm a full-time writer and I'm working very hard on *The Art of Computer Programming*. We have, uh ... let me just see if I can find a scratch page to work on ... [went to overhead with live Emacs screen on computer]. This is just to remind me about what to talk about ...

Now, it used to be that we used ACP as the abbreviation for *The Art of Computer Programming*. But someone else suggested that it should be called TAOCP. So now this is the new abbreviation for it: *The Art of Computer Programming*. This is my life's work, this is what I started working on in 1962, and I think I have about 20 years of work to go on it yet, after which I'll be 77 years old. So you see why I retired early — in order to be able to work very hard on this.

Bart Childs: Should that be TAOCP? [there was a typo on the screen, which showed TOACP]

DEK: It is. You want me to enlarge this font? I only have three or four fonts ... Oh! [sees typo]

Bart: Does that mean I get a check for five dollars and twelve cents? [laughter]

DEK: Not for you!! [laughter] ... You know I did that on purpose just to see if anybody was looking. [laughter] Alright ... Then there's this *other* book I'm working on, called *The Art of Computer Programming* [laughter] ... OK, so ... Boy, am I nervous. [laughter]

So, in order to finish this project, I have to work very hard, because computer science people keep discovering new things. Originally, my idea was that I was going to be able to summarize *all* of the good stuff in computer science, but now I have to say that I just have to work very hard in order to summarize all the *classical* good stuff in computer science. I'm working especially to get all the history correct and to lay the right foundation for the specialized things. But I can't go up to the frontiers in everything as I could have in the 60s, when I began the project. I worked on TEX for about 10 years total, I guess, and I'm hoping that those 10 years

[3] Wiles proved that $x^n + y^n = z^n$ is impossible when $n > 2$ and $xyz > 0$.

[4] Andrew Wiles, "Modular elliptic curves and Fermat's Last Theorem," *Annals of Mathematics* **142** (1995), 443–551.

[5] Maureen Schupsky, Managing Editor, Princeton University Press.

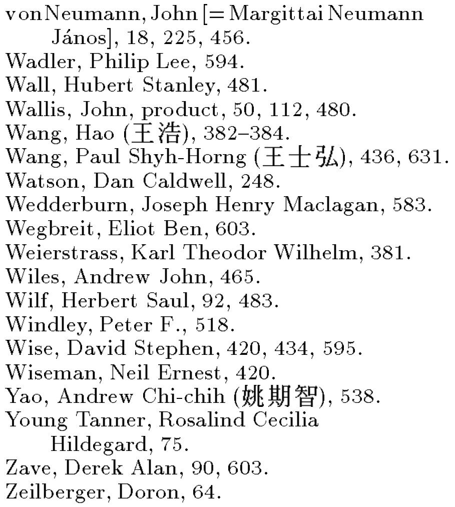
actually will save me about 6 or 7 years of the time I would have had to put into *The Art of Computer Programming* because I can now do my other work more efficiently.

The Art of Computer Programming is sort of what I view as the thing that I'm most uniquely able to do in my life. I'm feeling very healthy now and happy, and I feel that what I'm accomplishing every week is about as much as I've ever been able to do in my life in a week. So, I hope I can keep it up for a while. But I know that it takes a lot of time. That's why I'm retired and I'm working full-time on this.

I spent last year building infrastructure for the project, which meant making large computer files of what's in my house. So I have thousands and thousands of items that I've indexed and put into place so that I know how to find things.

Right now, my current project is to finish answering mail about *The Art of Computer Programming* that came in since I was working on TeX. [laughter] You know that if anybody found errors in *The TeXbook*, I answered the mail eventually and paid for the errors and so on. Well, people also get a reward for finding errors in *The Art of Computer Programming*. But the fact is, the last time I wrote a check for that was July of 1981. [laughter] In August of 1981, my secretary started issuing a form letter, typeset with TeX, saying "I will get back to you soon." [laughter] I started putting these letters into a little pile. Then the pile got to be a bigger pile, and it got mixed with all the other preprints I was receiving, until the pile grew to 260 inches high! Now, convert that to centimeters ... well, anyways, it's a lot! [laughter] It was about seven to eight meters of material. I went through all that and I am now answering those letters. Actually, the number of letters that I hadn't answered was less than 500 — something like 450 letters — and I'm now answering those letters and hoping that the checks will reach the people at the addresses from where they sent me their comments.

I'll show you the errata because I'm working on it now. Here's an example [Figure 1]. Just so you can see what the Index is about. This is p. 77 of the errata to Volume 1.[6] This is 8pt type being enlarged a lot. I wanted to show you one of the things I'm working on right now ... For all the authors that I cite in *The Art of Computer Programming*, when they have a non-Western name, I'm building a big database of the names in their native script, for example, Chinese or Japanese. (I haven't put in the Indian names yet, but I'm working with people in India to get that solved.) ... Right now, I have most up-to-date stuff on the Chinese part of it ... I have bitmap fonts for all the UNICODE characters — especially the Chinese characters — and I now have a pretty good database of these things, hopefully. So, by the time UNICODE software is ready and available, I'll be ready to use it and I'll be able to typeset the various names properly.



von Neumann, John [= Margittai Neumann
    János], 18, 225, 456.
Wadler, Philip Lee, 594.
Wall, Hubert Stanley, 481.
Wallis, John, product, 50, 112, 480.
Wang, Hao (王浩), 382–384.
Wang, Paul Shyh-Horng (王士弘), 436, 631.
Watson, Dan Caldwell, 248.
Wedderburn, Joseph Henry Maclagan, 583.
Wegbreit, Eliot Ben, 603.
Weierstrass, Karl Theodor Wilhelm, 381.
Wiles, Andrew John, 465.
Wilf, Herbert Saul, 92, 483.
Windley, Peter F., 518.
Wise, David Stephen, 420, 434, 595.
Wiseman, Neil Ernest, 420.
Yao, Andrew Chi-chih (姚期智), 538.
Young Tanner, Rosalind Cecilia
    Hildegard, 75.
Zave, Derek Alan, 90, 603.
Zeilberger, Doron, 64.

Figure 1: Excerpt from the end of `err1.dvi`

I have some interesting Emacs macros that help me with the UNICODE characters even though I don't have any software yet for UNICODE. I can type in four hexadecimal digits in Emacs and then say `M-X unic` and so magically it will convert that into the Chinese character — the bitmap of it — which then can be put into the document. There's Andy Yao's name in Chinese. I just have 24-by-24 bitmaps of all these characters, but it's enough for proofreading purposes. I have it set up so that it's very easy for me to get the characters into my file. The Emacs macro sends it to a little program that looks it up in a file, finds the bitmap, and inserts it into TeX format. Between angle brackets, the stuff's sent to PostScript. I'm planning that at the end of the year, I'll announce this errata list, which will be finished by that time. Right now, it's about 180 pages and I'm still building it while I'm answering these letters.

Then we're going to issue Volume 4 in fascicles, about 128 pages at a time. The idea is to do that about twice a year for the next 10 years. My steady state, I figure, is going to be about 256 pages a year of output. We're going to have three or four fascicles in hand before we actually start this publication. The first four fascicles — one of them will

[6] See Knuth's Web pages at `http://www-cs-faculty.stanford.edu/~knuth/taocp.html` –Ch.

be this larger one [the update errata to Volumes 1, 2 and 3]; the other three — the second one will be the design of a computer called MMIX[7] which is replacing the MIX computer. MMIX is a RISC machine, very much like the computers that we're all converging to these days. It's a 64-bit RISC computer which I might even *own* one day — I don't know, if somebody's going to build it. I've got the experts in the field helping me design it. Dick Sites, who was the architect of the Alpha chip, is one of my students and has promised to work on all the final steps of it. Also John Hennessy, who designed the MIPS chip. And some of the people from SPARC. So MMIX is going to be a nice clean RISC computer in order to make experiments on algorithms, to see how much the different cache management schemes work with different coding algorithms and so on.

The second fascicle will be to replace MIX by MMIX and I'm hoping that, eventually, every time I have a MIX program in *The Art of Computer Programming*, it will be replaced by an MMIX program. I'm not going to do that until I finish Volumes 4 and 5. But I'm hoping that a lot of other people will have done that work already by the time I get there. People have already promised me that they're going to have a C-compiler up for MMIX next year, and we're trying to get operating systems written for it.

The other two fascicles — one of them will be the first part of Volume 4, which talks about bit-fiddling. These are a collection of techniques that are mostly in folklore about efficient methods for computers where you're using the logical operations of a machine — the exclusive 'or', the 'and' and 'not', as well as 'plus', 'minus', 'times', and 'divide' — to gain great efficiency. I've got that material pretty much written already. In fact, it was what I had drafted just before I started working on TeX — that was 1977 when I wrote the first draft about bit-fiddling.

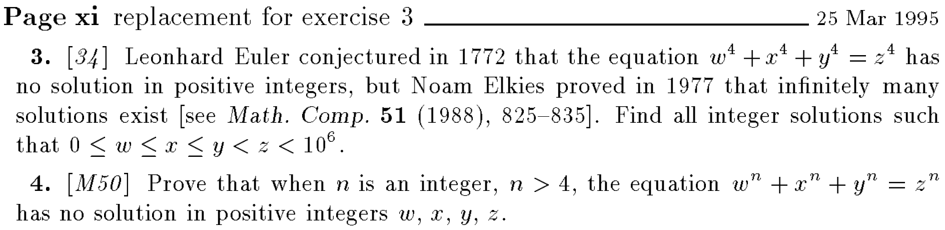
And then I get into the study of brute force enumeration methods. The subject of Volume 4 is combinatorial algorithms, and this means the methods that have been developed to deal with problems where you have zillions of cases; all kinds of ideas have come up as to how to speed up, by many orders of magnitude, the obvious methods for dealing with cases of combinatorial importance. I begin the chapter by talking about bit-fiddling, and the next part by talking about fast methods for listing all permutations, and listing all subsets of a set, and things like that. A vast literature about such things exists. Surprisingly, more people have written papers about generating permutations than about sorting. Sorting is the idea of putting into order; generating permutations is about putting into disorder. More people have explained how to unorder things than to order things. [laughter] Most of those papers are fairly repetitious and trivial, though, and not as interesting as the sorting papers, so the difficulty for me is mostly to survey this literature and put it all together. Most of the people writing on it were unaware that other people were working on the same thing.

OK. Well, that's more than enough of an answer to your question, I hope, on the state of Volume 4. Every day I should be able to finish about a page or so, and I think I've been going at about that rate for a while now.

While I've got this on the screen ... Let's try going forward a couple of pages ... I really want to look at this equation here [Figure 2]. I used to have Fermat's Last Theorem as a research problem at the beginning of the book. [laughter] In the errata it says now: "Prove that when n is an integer greater than 4, the equation $w^n + x^n + y^n = z^n$ has no solution in positive integers." So, now I've just added another variable to the equation [laughter] and we've got another good research problem. And it turns out that for $n = 4$, there are infinitely many solutions. The proof of Fermat's Last Theorem caused a personal crisis for me, but I've now resolved it in this way. [laughter]

I want to look at p. 61 as an example of new material [Figure 3]. This is about some very nice constructions that come out of studying trees. I put this up as an example of an illustration — I'm doing all the illustrations for the book in MetaPost, and this is one of the ones that happened to be around here. I love MetaPost, and John [Hobby], in the next talk this morning, will show you his system. It handles technical illustrations much better than anything else. The great beauty of MetaPost from my point of view is that if I have to modify any of my illustrations later on, maybe a year later, I can look at it and see from the MetaPost code exactly what I had in mind when I made the original. MetaPost is a declarative language where you state the characteristics that you want your illustration to have, and then it draws the diagram. I've got 5 or 6 examples in the errata where I've either redrawn an old figure or, as in this case, made a new one. Many, many other examples that I've done with MetaPost when I did the *Stanford GraphBase* book convinced me that it's really the answer for technical illustrations. There's nothing else I think

[7] MMIX = 2009, in roman numerals.



Page xi replacement for exercise 3 ———— 25 Mar 1995

3. [*34*] Leonhard Euler conjectured in 1772 that the equation $w^4 + x^4 + y^4 = z^4$ has no solution in positive integers, but Noam Elkies proved in 1977 that infinitely many solutions exist [see *Math. Comp.* **51** (1988), 825–835]. Find all integer solutions such that $0 \le w \le x \le y < z < 10^6$.

4. [*M50*] Prove that when n is an integer, $n > 4$, the equation $w^n + x^n + y^n = z^n$ has no solution in positive integers w, x, y, z.

Figure 2: Excerpt from the beginning of `err2.dvi`

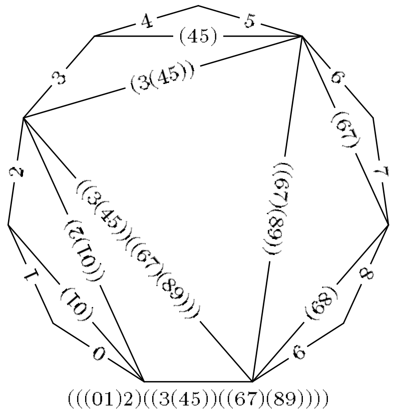


Figure 3: Excerpt from `err1.ps`, showing MetaPost illustration

will ever be able to be much better. It's not the answer for the kind of illustrations that people do when they're doing advertisements or something like that, but when you're writing a techical book, you have — the speaker yesterday made the same point with PSTricks — to produce a diagram that has a certain mathematical property to it. A Super MacDraw-type of program doesn't very easily give you this ability, while MetaPost does it, and very easily — all the technical things are correct, according to the mathematics. It's kind of scandalous that the world's calculus books, up until recent years, have never had a good picture of a cardioid. Yesterday, you saw a cardioid on the screen. Nobody ever knew what a cardioid looked like because it was done by some graphic artist who was trying to imitate another thing, but they never had a real one shown.

OK — I've talked too much about this. Ask another question!

Robert McGaffey: Do you think the RSA algorithm will ever be broken?

DEK: Do I think the RSA algorithm will ever be broken? Now, the RSA algorithm is the Rivest–Shamir–Adleman scheme for encryption. People have just factored the key of the original cipher that was put up by R, S and A in 1977. That was a 140-digit number, which was a very weak version of the thing. But Rivest et al. said: "Here's our secret message. Can you decode it?" By the time someone decoded it last year, Ron [Rivest] had lost the original answer ... forgotten what it was. But it turned out that, after decoding, the encoded message was: "The secret word is squeamish ossifrage." That was the solution. People, in order to break this cipher, had to factor a 140-digit number, and it was done with many thousands of hours of computer time last year. Now the thing is, though, if you go from a 140-digit number to a 141-digit number, already the problem gets much larger. So if you go to a 300-digit number, it would, as far as we know — all of the computers running now in the universe would not be able to do it. But there might still be advances in factoring, so Rivest himself predicts that a 300-digit encoding would last at least for about 30 years. A 500-digit number for a hundred years — he would rather confidently predict that's true.

We doubt if anyone's going to discover a magic way to factor numbers. The only problem is that it's illegal to use the full power of RSA. I mean, the government wants to be sure that it can read secrets if necessary, because it doesn't want the Mafia to have the secrets. So we now have a peculiar situation where it's against the law to compute a certain mathematical equation, a mathematical formula. Well, I don't like confrontational issues. [laughter] I don't live in a secret way ... I mean, I'm not a secretive kind of guy. I spent a year of my life working in cryptanalysis and I've met a lot of wonderful people in that community, but I knew that the life wasn't for me. I like to be a college professor and tell what I know. So, I'm not a good consultant on that kind of question. But it is possible to send secure information assuming that nobody can factor numbers. That's either a blessing or a threat, depending on your point of view. For me, I can see it from both sides.

In the back?

Michael Sofka: Is $P = NP$, and if not, how far do you think we are from the proof?

DEK: $P = NP$ is the most famous unsolved problem in computer science, analogous to Fermat's Last Theorem, although the $P = NP$ problem has only been around for about 30 years, 25 maybe. In the context of combinatorial algorithms, it says: Are we going to be able to solve problems that would require going through 2^n cases? Can we actually do those in n^{10}, or something like that, if we knew the best method? If $P = NP$, the answer would be 'yes', with some polynomial: we could reduce all these exponential problems to polynomial problems. If not, then it says 'no', we'll never be able to reduce them.

No, I have a feeling that someone might resolve the problem in the worst possible way, which is the following. Somebody will prove that P is equal to NP because there are only finitely many obstructions to it *not* being equal to NP. [laughter] The result would be that there is some polynomial such that we could solve all these problems in polynomial time. However, we won't know what the polynomial is, we just know that it exists. So maybe this will be n to the trillionth or something like that — but it'll be a polynomial. But we'll never be able to figure it out because it would probably take too long to find out what the polynomial is. But it does exist. Which means that the whole question $P = NP$ was the wrong question! [laughter] It might go that way. You see, even if you have something that takes 2^n steps and you compare it to something that takes n^{100}, then at least you can solve the 2^n one for n up to 20 or 30. But the n^{100} you can't even do for $n = 2$. So the degree of that polynomial is very important. There are so many algorithms out there, the task of showing that no polynomial ones exist is going to be very hard. Still, I really thought with Fermat's Theorem, it was a similar kind of thing, where it was more important to have the problem than to solve it. Therefore, my real feeling about Wiles's Theorem is that he did a marvellous wonderful piece of work, but I wish he'd solved something else! [laughter]

A lot of people think that as soon as a problem is shown to be in this class NP, they shouldn't work on it, because it means that there's probably no polynomial way to solve the problem. But before we studied NP, we had *unsolvable* problems, problems for which there didn't exist any algorithms at all — no matter how long you worked, you would never solve the problem. To tell whether a Turing machine ever stops: this is an unsolvable problem by any algorithm, no matter how long you give yourself. So, people would stop working on a problem as soon as it was proved to be unsolvable in general. But that was a bad strategy, because almost every problem we ever solve is a special case of some unsolvable problem.

Take calculus, for example. The problem of taking a formula, a function of n, and saying: "Is the limit as n goes to infinity equal to zero or not?" That's an unsolvable problem. But its unsolvability doesn't imply that we shouldn't study calculus. I mean, limits of lots of functions we've been able to decide do go to zero, and therefore people were able to develop calculus. But it's an unsolvable problem. I mean, you could define f of n — it only takes a few lines to make a formula that is equal to zero if a given Turing machine is stopped at time n, and it's equal to 1 if the Turing machine is still going at time n. And so the limit is equal to zero if and only if that Turing machine stops. It's unsolvable.

A similar thing happens with NP. That is, we have lots of special cases of problems that are NP-hard that we can solve efficiently; just knowing that something is NP doesn't mean that it's a good idea to give up on it or to stop trying to get good heuristic methods for it.

Questions about TEX?! [laughter] Yes?

Raman: One of the nice thing about TEX is that it gives the authors the flexibility to define macros that sort of encode semantics, so if you're writing a paper about permutations you can define a thing called `permute` and then write your contents using that. I rely heavily on this in my system ASTER. What I wanted to know —

DEK: Excuse me, you're going to be talking later on ... ?

Raman: Yes, I'll be talking tomorrow afternoon.

DEK: So your system uses the TEX source as part of the semantics of the presentation of the document, while the author is thinking mostly of the convenience of writing.

Raman: Yes. So, what happens, in fact, is that everything turns into an object. And so, the more semantics there are in the mark-up, the better it is. Now, in normal documents, there is this tension between wanting to write things using a base level of mark-up where an author just writes `x backslash over y` whenever he wants 'x horizontal rule y' versus an author defining, say, `\inference` as a macro that takes two things and then puts x over y. In the latter case, I win; in the earlier case I lose. My gut feeling is that, if you look at a large book, it's like a large computer program, and, sort of, in order to preserve your sanity, it seems to make more sense to read it that way.

DEK: Can people hear what the question is? He's saying that if you're blind or handicapped, you can look at the TEX source of a document and if it's properly done, it can even be better than if you had the hardcopy, because the document could have been written with a very logical mark-up scheme. In typography we try to reveal the structure by typographic means. But, in fact, we know even more of the structure when we're making our source files. So, an author with that in mind would maybe prepare the source files to have more information than just what you're going to see on the page afterwards.

When you're writing, you have an audience in mind. If you look at any book about how to write, or any course that deals with writing, the number one rule they have is: Keep your reader in mind. If an author realizes that he's writing something for hypertext, then he'll be planning his exposition so that it takes best advantage of hypertext. When I wrote my first paper in a foreign language it was published in Canada, where they spell 'color' with a 'u'. [laughter] My second paper in a foreign language was Norwegian, and so I when I was talking about variables for 'left' and 'right', I would say 'h' and 'v' instead of 'l' and 'r'. I mean, you plan as you're writing, you plan for the reader. These are very simple examples. So, if you expect that somebody is looking at your TEX source or that somebody will be able to click on part of your document and therefore it'll highlight something that is logically related, you might approach the whole process of exposition in a different way and you'll be able to reach more readers. So I try to make my … I can show you the macro files developing for *The Art of Computer Programming* in the new style, but …

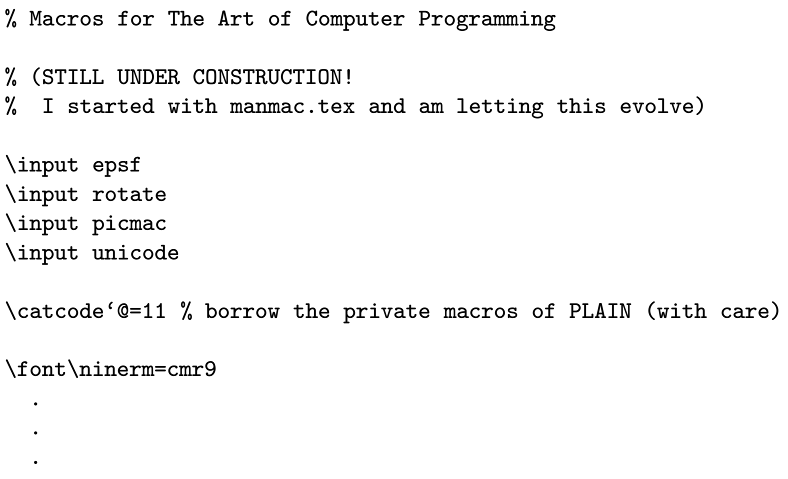
Raman: I'd love to see that, because the other thing I'd like to do is run the system off the …

DEK: [more screen manoeuvers; see Figure 4.] Let's just take a look at that file. It's still under construction, ok? [laughter] I input various macro files: the "`epsf`" is to get the MetaPost figures; "`rotate`" allows PostScript to do rotation; "`picmac`" is my subset of LATEX picture mode; "`unicode`" is that thing that I told you about for getting Chinese names. [goes through the macro on the screen — not transcribed] … I type `6709` and then I say `M-X-unic` and magically this appears on my screen — that's my temporary solution to the UNICODE problem. OK, now. Let's take a look at some formatting … [more pointing out of various codes for formatting]

These are my composition macros [Figure 5] … some of the hyphenation exceptions I've put in; equation numbers are going into oldstyle numerals; star is for a starred section, something that's advanced; slug means at the end of a proof — I might have to change it so it's not so black, because people see so many overfull boxes, they don't like to see the black slug anymore. [laughter] I'm redefining backslash dot — I make it a colon so I can use `\.` for typewriter type in the middle of math mode. There's an important notation due to Iverson, where you can put any formula in square brackets and that evaluates the formula to zero or 1 — you can use that in the middle of equations, it's very useful. Here are macros for saying A.D. or B.C. Here's something for special emphasis I use, like a hint — I'm not using this to indicate italics, I say `\it` for that kind of emphasis. My `\em#1:` is a special format that I often use for a hint or a note or something like that [walking through the macros, with comments on some of the more interesting ones] … These are Eulerian numbers … with angle brackets instead of the parentheses of binomial coefficients.

Raman: Do you have a macro called `\euler` there?

DEK: Yeah, these are Eulerian numbers here, yeah. … Capital `\Euler` is where it uses two delimiters and has the right amount of negative space between them so that you have two angle brackets next to each other; similarly, for binomial coefficients, the `\Choose` puts in two parentheses. Here's `\smallsum` for a small summation sign — I don't remember where I used it. `\phihat` — this is a special symbol — the letter ϕ has to have the hat put just right, because it's a common thing that arises when you're studying Fibonacci numbers. … Then I have here the format for an algorithm […]; a whole bunch of macros for typesetting assembly code; underlining text in the comments … [remaining examples not included]



```
% Macros for The Art of Computer Programming

% (STILL UNDER CONSTRUCTION!
%  I started with manmac.tex and am letting this evolve)

\input epsf
\input rotate
\input picmac
\input unicode

\catcode`@=11 % borrow the private macros of PLAIN (with care)

\font\ninerm=cmr9
  .
  .
  .
```

Figure 4: The beginning of `acpmac.tex`

Right now, my TEX file has some structure of that form. Let's take a look at the file itself:[8] This is where I'm working day-by-day to put in my new corrections — unfortunately, we have a real narrow screen here. I have four kinds of errata: One is called an 'amendment', which is something new, that we didn't have before; one is called a 'bug', something that has to be fixed; one is called a 'plan', which is something where I'll work out the details later but I want to note down in the file that it's in my mind to make the change; and then there's something called an 'improvement', which is kind of trivial, but still I thought of it and I want to use it when we go to the final book.

For example [Figure 6], here's a quotation by Turing I kind of like from 1945, which is before computers were even invented, but he'd been thinking about it for a long time: "Up to a point, it's better to let the snags be there than to spend such time in design that there are none. How many decades would this course take?" That quotation is an amendment to Volume 1.

Look, here are the amendments for Volume 2, p. 2 [Figure 7], and I made this change on July 13; you can see these are the things I'm working on right now. Here's a new exercise I put in, here's an example where I changed a comma to a semi-colon. The way I use this is that trivial improvements don't usually get listed in the hardcopy unless you work extra hard. There's a special way of getting all these improvements to come out, but usually they only appear in the file.

So that's the kind of thing I'm doing. This improvement, by the way, is dated 1981. For 20 years, I've had copies of *The Art of Computer Programming* sitting in my office and I kept putting notes in the margins, marking things that I want to improve. That's now all in these files.

Other questions?

Silvio Levy: How come you don't use LATEX? [laughter]

DEK: How come I don't use LATEX? [laughter] I'm scared of large systems! [louder laughter] Bart?

Bart: Your paper, "The Errors of TEX," was great. Have you ever thought of one about "Mistakes of TEX"?

DEK: "The *Mistakes* of TEX"?? [laughter]

Bart: I mean, I guess I'm kind of thinking of the changes you made when you went to TEX3. The 7-bit/8-bit and things there that might be thought of as mistakes. Are there any other things you can think of in that line?

DEK: In that paper, I think I listed everything that I would consider a mistake. I think I would have noted that [reference to 7-bit/8-bit], but of course, I wrote that paper before TEX3 came out. I've promised to put out a sequel to that paper when everything has cooled down and you know, when the last error in TEX has been found. [laughter] So the present state is this. [File manipulations on screen bringing up the file `errorlog.tex` (which is in the CTAN archives under `system/knuth/errata`); see Figure 8.] The last change was on March 19. Well, no, that was the date the bug was reported. So, here's Peter Breitenlohner — he's here today — "Avoid spurious reference counts in format files".

[8] This macro file `acpmac.tex` is downloadable from the web page mentioned earlier.

```
  .
  .
  .
% Composition macros

\hyphenation{logical Mac-Mahon hyper-geo-metric hyper-geo-met-rics Ber-noulli
 Greg-ory dis-trib-uted}

{\obeyspaces\gdef {\ }}
\def\hang{\hangindent\parindent}
\def\dash---{\thinspace---\hskip.16667em\relax}
\def\eq(#1){{\rm({\oldsty#1})}}
\let\EQNO=\eqno \def\eqno(#1){\EQNO\hbox{\eq(#1)}}
\def\star{\llap{*}}
\def\slug{\hbox{\kern1.5pt\vrule width2.5pt height6pt depth1.5pt\kern1.5pt}}
\let\:=\. % preserve a way to get the dot accent
\def\.#1{\leavevmode\hbox{\tt#1}}
\def\[#1]{[\hbox{$\mskip1mu\thickmuskip=\thinmuskip#1\mskip1mu$}]} % Iverson
\def\bigi[#1]{\bigl[\begingroup\mskip1mu\thickmuskip=\thinmuskip
          #1\mskip1mu\endgroup\bigr]} % big Iverson brackets
\def\AD{{\adbcfont A.D.}}
\def\BC{{\adbcfont B.C.}}
\def\og#1{\leavevmode\vtop{\baselineskip\z@skip \lineskip-.2ex
  \lineskiplimit\z@ \ialign{##\cr\relax#1\cr
    \hidewidth\kern.3em\sh@ft{40}`\hidewidth\cr}\kern-1ex}} % ogonek
\def\em#1:{{\it#1:\/}} % \em Hint: or \em Caution: or \em Reference: etc
   .
   .
   .
\def\euler{\atopwithdelims<>}
\def\Euler#1#2{\mathchoice{\biggl<\mskip-7mu{#1\euler#2}\mskip-7mu\biggr>}%
 {\left<\!{#1\euler#2}\!\right>}{}{}}
\def\Choose#1#2{\mathchoice{\biggl(\mskip-7mu{#1\chooser#2}\mskip-7mu\biggr)}%
 {\left(\!{#1\choose#2}\!\right)}{}{}}
\def\smsum{\mathop{\vcenter{\hbox{\tenrm\char6}}}} % small summation sign
\def\phihat{{\mkern5mu\hat{\vrule width0pt height1.2ex\smash{\mkern-5mu\phi}}}}
\def\umod{\nonscript\mskip-\medmuskip\mkern5mu% least remainder (underline mod)
  \mathbin{\underline{\rm mod}}\penalty900\mkern5mu\nonscript\mskip-\medmuskip}
   .
   .
   .
```

Figure 5: Excerpt from acpmac.tex

```
\amendpage 1.189 insert quotation before the exercises (95.07.13)
{\quoteformat
\vskip-3pt
Up to a point it is better to let the snags [bugs] be there
than to spend such time in design that there are none
(how many decades would this course take?).
\author A. M. TURING, Proposals for ACE (1945)
% p18, quoted in Comp J 20(1977)273
}
\endchange
```

Figure 6: One of the errata in err1.tex

```
\amendpage 2.2 line $-2$ (95.07.13)
digits, and in 1955 \becomes digits. The Ferranti Mark~I computer, first
installed in 1951, had a built-in instruction that put 20 random bits
into the accumulator using a resistance noise generator; this feature had
been recommended by A.~M. Turing. In 1955,
%  "resistance noise generator" -- quoted from Brooker's manual for Mark II
\endchange
```

Figure 7: One of the errata in `err2.tex`

```
  .
  .
  .
* 26 June 1993
R928\>668. Avoid potential future bug (Peter Breitenlohner). @628,637
* 17 December 1993
S929\>881. Boundary character representation shouldn't depend on font
 memory size (Berthold Horn). @549,1323
* 10 March 1994
R930. Huge font parameter number may exceed array bound (CET). @549
* 4 September 1994
F931\>926. Math kerns are explicit (Walter Carlip). @717
R932. Avoid overflow on huge real-to-integer conversion. @625,634
* 19 March 1995
R933. Avoid spurious reference counts in format files (PB). @1335
\relax
\bye
```

Figure 8: End of the file `errorlog.tex`

This was causing some problems ... he found you could break TeX if you kept saving format files and loading them again and saving them again and loading them again several hundred times; you would exceed memory capacity because the reference count could get larger than the total memory size. So that was a bug that we fixed, and he got $327.68 for that — 2^{15} pennies.

What were the other most recent changes? Number 932, "Avoid overflow on huge real-to-integer conversion." Number 931, "Math kerns are explicit" — this was a bug introduced by change 926. Number 930, "Huge font parameter number may exceed array bound," a place where the implementation wasn't totally robust. Number 929, "Boundary character representation shouldn't depend on font memory size" — this was a fairly serious one that was fixed by the major implementors shortly after we put out the previous update in 1993.

These errors — the dates listed here are actually dates when the people found them. I fixed them all in March of this year [1995]. I plan to look again at reported bugs in TeX in 1997, and again in 2000, and then 2004 and 2009, hoping that each time I'll be able to do that in about a day. There are a lot of people out there filtering these reported bugs, and vetting that they really do seem to require my attention. Yesterday, when we were running some programs at Stanford, somebody noticed that Stanford was still using a very old version of TeX and it didn't seem to matter. [laughter] I believe the bugs are starting to taper off. The remaining ones are getting to be scenarios that can cause it to break, but only if you really try hard.

There's one severe bug in the design that will have to remain as a feature, and it has to do with multilingual typesetting. I don't think I put it in the `errorlog` file, but it's noted at the end of another file called `tex82.bug` [Figure 9]. Let's look there ... this file has complete details about every change since 1982. See this one? It's the "absolutely final change to TeX, to be made after my death." The version number changes to `$\pi$`. It's like my last will and testament here. [laughter] So I'll never see that change made. That's when TeX is declared to have no more bugs. Anything that uses that name should be fully compatible with everything else that uses that name.

After that final change, this file lists "possibly good ideas that I won't implement"; ... then come two "design errors that are too late to fix." The most serious one is multilingual. If you're using several

```
-----------
415. The absolutely final change (to be made after my death)
@x module 2
@d banner=='This is TeX, Version 3.14159' {printed when \TeX\ starts}
@y
@d banner=='This is TeX, Version $\pi$' {printed when \TeX\ starts}
@z
When this change is made, the corresponding line should be changed in
Volume B, and also on page 23 of The TeXbook.
My last will and testament for TeX is that no further changes be made
under any circumstances. Improved systems should not be called simply `TeX';
that name, unqualified, should refer only to the program for which I have
taken personal responsibility.  -- Don Knuth

* Possibly nice ideas that will not be implemented

. classes of marks analogous to classes of insertions
  .
  .
  .

* Design errors that are too late to fix

. additional parameters should be in symbol fonts to govern the space between
  rules and text in \over, \sqrt, etc,

. multilingual typesetting doesn't work properly when the \lccode changes
  within a paragraph

* Bad ideas that will not be implemented

. several people want to be able to remove arbitrary elements of lists,
   but that must never be done because some of those elements (e.g. kerns
   for accents) depend on floating point arithmetic

. if anybody wants letter spacing desperately they should put it in their own
   private version (e.g. generalize the hpack routine) and NOT call it TeX.
```

Figure 9: End of the file `tex82.bug`

languages in the same paragraph, there was some part of the state information that I forgot to save. I forget exactly what it is now,[9] but it was a serious oversight and I should have thought about it, but now it's too late. So that's a glitch that's going to have to remain.

The only other real thing I wished I'd worked harder on was the positioning of square root signs and fraction bars. I don't have enough parameters in there to control the space between the barline and the text. I made the mistake of solving a problem where I needed two parameters by using only one parameter: I got the amount of space by calculating it as a multiple of the thickness of the barline. And I should have had two parameters. Now I find that as I'm writing stuff and I have a square root that doesn't look right, I have to put a hidden strut in the exponent, to give more space there. I wish I'd done that better, but otherwise, considering the amount of inevitable compromise that has to go into any large system, I'm basically happy with the way things converged.

As I read papers typeset with TEX, the main thing that makes me unhappy, besides the way I typeset the square root sign, is the way that people have not updated to the Computer Modern fonts that I put out three or four years ago. We're still seeing the old fonts and I don't know how long it's going to take before people change. Eberhard [Mattes] did make the switch two or three months ago. It's especially evident on lowercase Greek delta.

[9] They all have to use the same `\lccode` table.

I found myself four or five years ago writing a paper and I found myself *not* using the letter delta in the paper. I tried to analyse, "why am I not using the letter delta?" and I realised that I subconsciously hated my letter delta; I didn't like the look of it. But in this paper I really needed the delta, it was the natural letter to use, and so I said, "OK ... I'm gonna take a day and redesign it and make a really beautiful delta." And I think I now have the best delta the human race has ever seen. [laughter] That was ages ago. Now, every time I see a paper using the old one, I cringe.

I also changed a few of the other letters, like some of the calligraphic capitals. I fixed the spacing on the barline on the 'H', and I didn't like the base of the 'T'. I see the NTG uses the old one in their logo ... but I'm hoping everyone will switch over to the real CMR fonts. The .tfm files have not changed. Oh, I've also made all the arrows heavier. The arrows were disappearing on xerox all over the place, and now the arrowheads are darker.[10]

Cameron Smith: Is there a date or version number we should look for to make sure ... ?

DEK: Well, if it's '93 or after, it'll be ok.

Silvio: No! Change CM to DM! It's never going to happen unless you do that.

DEK: It's happened in most places by now, but there still are pockets of people who haven't upgraded the old files. If you use dvips, all you have to do is delete the .pk files and it'll make them all for you.

Silvio: If you've got the new version of the source.

DEK: Yeah, well, the sources are all there.

Silvio: Right, but it's very hard for the public to ...

DEK: Please, figure out a way to solve this, because it's frustrating. Every time I get a letter from someone that has the old delta, I just tell them to tell their computer operators to update, and then they send me back an upgraded paper and say, yes it's ok now. [laughter] As long as people are aware of it. It's not that much of a change. If we get the word around to the distributors and the math journals, it shouldn't take too long to converge to that.

Robin Fairbairns: Can I just make a comment on that? Eberhard Mattes is possibly the author of the version with the largest user base — he has just reissued everything, and a month back he produced an entirely new set of fonts. On the mailing list there is a continual whinging about "we don't want to go to the trouble of updating our font files." I keep saying "You really *do* need to do this." But despite that, people say, "It costs computer time on our PCs."

DEK: They'll get a new PC in five years. [laughter] It'll eventually happen. I'm just hoping that it will happen sooner rather than later.

Silvio: Look, I posted a copy for people in Australia. I can tell you the top priority will not be to update because of a delta. If you issue a new version, with its own number and a new name, and if you make it obligatory, otherwise, ...

DEK: It didn't change that much to make it obligatory.

Jeremy Gibbons: If you change the name, old .dvi files won't work when you print them — there'll be lots of missing fonts.

DEK: But it's not that important. I mean, I'm too much of a nit-picker; I'm just telling you it does offend me, but it apparently doesn't offend those other people. [laughter]

Silvio: It offends me, too!

DEK: Oh, OK. Well, I don't want to change the names of my fonts. [pause] Nelson?

Nelson Beebe: Don, the world has changed a lot since 1978 —

DEK: Yes, in fact I put a wonderful quote from Bill Gates that said exactly the same thing at the beginning of my errata list this year ... [Figure 10].

Nelson: Assuming you were 25 years younger and were sitting down to do TeX now, with the market full of word processers and PostScript laser printers and so on — What would you do differently?

DEK: Well, as far as I know, I would still do the same thing, pretty much. So, anything that you don't like, I'd probably still put in! [laughter] It's just the way I do things <laughing>.

Cameron: There's a particular point, related to that, that I wanted to ask you about. You went to a lot of trouble to design a line-breaking paragraphing algorithm that looked over a wide range of possibilities for an optimal set of breaks. But I encountered in *The TeXbook* that computer memories being what they were, it wasn't practical to similarly accumulate several pages of material and look for optimal page breaks. And sort of related to that, there's the difficulty of communications between a line-breaking algorithm and a page-breaking algorithm, where, let's say you're doing a letter and

[10] Knuth has now put a discussion of this issue onto one of his Internet web pages: see http://www-cs-faculty.stanford.edu/~knuth/cm.html –Ch.

```
\vfill
{\quoteformat
Things have changed in the past two decades.
\author BILL GATES (1995)
%% CTdbl quote marks
% ~You, too, can start a software firm"
% International Herald Tribune 5 Jan 1995, pages 9 and 11
\bigskip
In addition to the errors listed here,
about half of the occurrences of ‘which’ in volumes one and three
should be changed to ‘that’.
\author DONALD E. KNUTH ({\sl The Art of Computer Programming Errara %
                             et Addenda}, 1981)
\eject
}
```

Figure 10: Another excerpt from `err1.tex`

there's a letterhead on the first page that forces you to have a different page width and you might need to have line breaks change in the middle of paragraphs. Things like that that could have been simplified if there were the ability to defer the cut-off of the page and have better communications between the line-breaking and page-breaking algorithms. Would you redo something like that, now that we have multi-megabyte memory?

DEK: OK, certainly the memory constraints are quite different now. Amazing how much — memory has changed more than anything else. There are also major changes in the way we — well, we've got many more years of experience. We understand these things now. At the time when I was working on TeX — I'm trying to put things in context — many things were experimental, so that we could learn about the territory. In any system design, whenever you go through a new generation, it turns out that you understand the previous generation and you clean up the previous generation, and then you also go into your new experiments, which have to be cleaned up by the next generation. That kind of traditional growth of understanding is the way the world works.

Now a lot of these things about what kind of communication would be useful and so on are becoming clearer. The idea of TeX was — and I think will remain for as long as it survives — to find the smallest number of primitives that would be able to handle the most important things. So that 99% of the work would be done by these primitives, and the other part would be done by tinkering. My attitude on these things is that when I have a job to get done, I don't ever expect to have a system that's going to do 100% of it for me, but I expect it'll do so much of it so that the tinkering is down to noise level. It's no more than a small percentage of the time I've put into the other parts of the job. Noise level for one person is different from another.

For example, I did this book about the Bible[11] where I had a lot of illustrations. I spent 6 or 7 hours on each illustration, preparing it, including color separations, fat-bits editing to clean up joins and various things, some scanning. For me, that was noise level, because I had already spent 40 hours of work writing the chapter that goes with the illustrations — so what's another 6 hours, I mean, it's a small percentage of the job, in some ways.

But if I'm going through a commercial place that's trying to get graphics in and out the door, somebody's paying for their work, and if I have 60 illustrations taking 6 hours each, that's a completely different game. So, my view is that different users will have a different idea as to how much really has to be automated. Many things are relatively easy for an author to spend a little extra time doing, because he's already put much more time into writing the book. But people who work with the author might want them to be automated and part of a fancy system the author doesn't want to take time to learn.

So, if I have a typographic task where I need to do something in 2 or 3 passes, well, I'll just try it a couple of times and run it through the machine and look through the previewer and get it right. A week or two ago, we put out a newsletter for our family. My wife does this every year. We have four grandparents and each of them has an extended family of,

[11] Donald E. Knuth, *3:16 Bible Texts Illuminated* (Madison, Wisconsin: A-R Editions, 1990); reviewed in *TUGboat* 12, #2 (1991), 233–35.

I don't know, about 60 or 80 names. We write to them and say, "Would you like to send us your comments on this year?", and then we collect them all together in a little booklet and send it out to four groups. So I fiddle with that for, I don't know, 6 to 8 hours just to do neat things; like a newsletter editor. ... You spend some time doing all kinds of prettying up, if you have the time to put in. The tools you have change the expectations of what you try to accomplish.

So, I figure the next generation of systems will have a lot more complicated mechanisms to handle the general cases of everything, where I've considered only the cases that I thought were the 99%. There are all kinds of complexities — maybe you want to have the reference point in the middle of a character instead of at the left edge, as you're going left and right, in color and rotated and in many columns, and so on. We now have more understanding of how to design such general mechanisms.

With respect to the memory situation ... I think the page-breaking business is still ... it's not so much memory-bound as maybe — you still want to do 2 passes, but the machines are fast enough for two passes — it isn't that slow anymore. So, I would say this kind of next generation thing is natural for other people to work on, with the understanding that they gained from the first system.

Pierre?

Pierre MacKay: Just going back to the question of upgrading the fonts. I was thinking about something that wouldn't break things. You stick a METAFONT `\special` to identify the font version. Since people upgrade their drivers far more often than they upgrade their fonts, just have the driver recognize that `\special` and say "Tut tut! You shouldn't be using this font!" [laughter]

Fred Bartlett: I think that everything the gentleman over here wants to do with line-breaking and page-breaking could be done fairly simply by writing moderately complex macros and new output routines, *if* there were a way to save the items that get discarded at the end of every page. You want to save the discardable items that get tossed out when TEX calls the output routine. I was wondering why, when you wrote TEX, you threw those away without making it possible to save them at all?

DEK: The output routine can put it into a box — copy it into a box —

Fred: — but it can't save the skip that is thrown out.

DEK: The skip that's thrown out, isn't that the value of one of the parameters that gets passed to the output routine?

Peter Breitenlohner: The penalty is passed, but the skip is thrown out after the page has been printed. And if nothing comes back, it is not thrown out.

DEK: Somewhere in *The TEXbook* it gives a null output routine that's supposed to put everything back together again? And what — that doesn't work?

Peter: It works! Because then the skip is not at the top.

Fred: But you can't ship out a page, you can't save a page to a box and then go back, accumulate a new page and then push it through and save that to a second box, as for left and right-column setting, and then put the two boxes together and have them join smoothly, because then the skip in between will be missing, and you'll have ... you won't have the line skip in between, and it means that if you want to do complex 2-column setting, as for a couple of text books [I've done], then you have a problem with tables and figures and a whole bunch of other junk. It would be nice ... it's something I've wrestled with for, I don't know, 6 or 7 years, and the best I've been able to do is to have TEX warn me when it starts balancing columns, starting on the right-hand column, and because I probably don't have the right skip here. Almost everything else in TEX is parameterized: you can get to it, you can save it, you can inspect it, have TEX do tests — except the discardable items, and discardable skips. I'm just wondering why.

DEK: I guess I didn't think of it. [laughter] The output routine was the most experimental part of TEX. We had no models to go by at all. We had 4 or 5 problems that we knew we had to solve, and we tried to find the smallest number of primitives that would handle those 4 or 5 problems. We got to the point where we could clarify the solutions to the problems. But we *knew* this was experimental. I'm sorry that your problem didn't occur before TEX 3.0 because then I might have been tempted ... [laughter]

Fred: I heard you say you expected more people to extend TEX than have done so.

DEK: Yeah, absolutely. I expected extensions whenever someone had a special-purpose important project, like the *Encyclopedia Britannica*, or making an Arabic–Chinese dictionary, or whatever — a large project. I never expected that one tool would be able to handle everybody's exotic projects. So I built a lot of hooks into the code so that it should be fairly easy for a computer science graduate to set up a new program for special occasions in a week or so. That was my thought. But I don't think people have done that very much.

It's certainly what I would have done! If I were putting out a Bible or something, if I were a publisher with some project that I wanted to do specially well, then I would want a special typesetting tool for it. Rewriting a typesetting system is fairly easy. [laughter]

I guess people haven't done it because they're afraid they'd break something. I don't think they would have. I think the caution is misplaced. So I tried to show how to do it, by implementing several of the features of TEX as if they *were* added on after, just to show how to use the hooks, as a demo. But that didn't get things going. So, many more people are working with TEX at the macro level. Of course, the big advantage is that then you can share your output with others — you can assume it's going to work on everybody else's systems. But still, I thought special projects would lead to a lot of custom versions of the program. That hasn't happened.

Jeremy: A related question is ... if you can take a vertical box apart into its components, and play with them and reassemble it, one thing you *can't* get is, if you have a box that has been moved left or moved right, when you disassemble the box, you lose that information. [**DEK:** Oh really?] You can get `\lastbox`, and that gives you the box, but it doesn't tell you whether it was shifted. I thought I saw in your list of bad ideas that weren't going to have anything done with them, something to do with taking lists apart. Was that there?

DEK: I'm not sure what that really referred to any more. There is some problem about making sure that no user can access the results of rounding errors, which are different on different machines. So I had to be very careful in TEX to keep it portable — any time you do a glue calculation. Still, I don't think that would happen in shift-left shift-right. One of the changes to TEX not so many years ago was to ... I don't remember. Maybe somebody can ... maybe Peter [Breitenlohner] can recall. I think it was you who suggested it: There was something in the *hlistout* and *vlistout* routines where it looked at the shift amount of the box and ... ?

Peter: It was in leaders. The leaderbox looked for the shift amount, but the shift amount in the data structure was always zero ... TEX took the shift amount, and added it and subtracted it back, or something like that.

DEK: Yeah, so the thing is, I had some code in there that — it wasn't a bug because it could never cause any harm — but I was always adding zero to something. We took it out, so that people wouldn't be confused by it. The amount by which a box is shifted is stored with the box. If it's in a vertical list, that means so much is shifted towards the right and in a horizontal list, it means how much it's shifted up and down. That could never be nonzero in a leaders box. OK, if the shift amount is not restored properly, it might be a bug, something you could report, and in 1997, maybe you can get big money for it. [laughter]

Jeremy: I don't think it's a bug, it's just a problem, something you can't do — taking things apart and reassembling them ... It wasn't a design decision ... a deliberate one?

DEK: Alright, well, the number of possible things like that was too huge to anticipate, so I just am glad that there weren't more, I guess. I'm sorry.

Cameron: A lot of the questions have been of the form "Why did you do X?" but I think maybe part of what the thrust really is, is: If you were doing it again, as some people are trying to do, and as you've suggested that more people *should* be doing, reading from TEX for special needs, are there things that you'd recommend to *those* people? Not so much why did you do it this way 20 years ago, but if someone else were doing it again what would you tell them is most important to think about?

DEK: I just recommend putting extreme care into the design and checking things out and getting a wide number of users to help you with it, to show you the problems that they have and look at as many examples as possible. These are the things people are of course doing already. Dotting the I's and crossing the T's is the name of the game; you have to work very hard over a long period of time. The more you open or extend your vision as to how much you're going to solve, the longer it's going to take to get the whole package to be consistent.

The hardest struggle is the struggle towards convergence, instead of divergence. You need input from a tremendous variety of sources, but you also have to avoid the committee ... you have to have some small number of people who makes the decisions, in some way, so that it converges. Otherwise, you get the big problem of all committee projects — that everybody on the committee has to be proud of one part of the final thing. Then you have lots of incompatible stuff in there, mostly for political reasons. The hard thing is to do the detailed checking on as many things as you can for consistency and convergence. The steering problem is the hardest thing with TEX.

If you study the paper, "The Errors of TEX," you'll see how this worked.[12] First there was one user — and I took a lot of time to satisfy myself. Then I had 10 users, and a whole new level of difficulties arose. Then I had a hundred users and another level of things happened. I had a thousand users, I had ten thousand — each of those were special phases in the development, important. I couldn't have gone with ten thousand until I'd done it with a thousand. But each time a new wave of changes came along, the idea was to have TEX get better, and not get more diverse as it needed to handle new things. So, when I said I'd still do things pretty much the same way, what I meant is I still think I would have horizontal lists and vertical lists; I still would have boxes and glue, and so on. That basic structure seems to give a lot of mileage from a the small number of concepts, to handle a tremendous variety of typesetting challenges. But I wasn't talking about whether I would do exactly the same with respect to a shift amount here and there. All those fine points are extremely important, but I'd still keep the same basic architecture.

Barbara: It's getting on to refreshment time. So, I would like to thank Don very much for taking time to answer everybody's questions. I will take the prerogative of one last question: Would you be willing to do this again in 2011?! In 16 more years?

DEK: Yeah, that sounds about right. [laughter] Thirty-two isn't quite as nice, but it should be OK. [laughter] Thank you very much. [wide applause][13]

[12] "The Errors of TEX," *Software Practice & Experience* **19** (1989), 607–785; reprinted with corrections and additional material as Chapters 10 and 11 of Knuth's book *Literate Programming*, distributed by Cambridge University Press. See also "Notes on the Errors of TEX," *TUGboat* 10, #4 (1989), 529–31, the keynote address at the 10th Anniversary Meeting, held at Stanford in July, 1989.

[13] As the keyboarding transcriber for this talk — arranged as the audience was just starting to sit down — I have to drag in some names of people who'd had a lot more foresight than I — they had brought mini-cassette recorders along just for the occasion! It was Cal Jackson who generously offered me his tape right after the talk had concluded. Several months later, it was Jeremy Gibbons who casually mentioned in e-mail that he too had taped the talk and would be quite happy to check my transcript. Many of the gaps which I had been unable to decipher were taken care of by Jeremy's super recording. And finally, I must of course thank Prof. Knuth, who graciously took time to read over the edited version to ensure accuracy. –Christina Thiele

Software & Tools

CAMEL: Kicking over the bibliographic traces in LaTeX

Frank G. Bennett, Jr.

1 Introduction

It gives me great pleasure to introduce readers to a prototype citation manager called CAMEL. This package, as it now appears on CTAN, is work in progress of the LaTeX3 bibliography team. The implicit goal of this project is as simple as it is precocious: to create a new citation interface for LaTeX that is simple, intuitively straightforward, and allows the bibliographic style of a document to be changed at will without major editing. In effect, we are seeking to implement logical markup for citation styles, cross-referencing schemes, and bibliography formats. If the savings in editorial time that this implies are of interest to you, read on.

While the long-term goal of our work is to produce a fresh standard bibliography package that will be widely used and well supported, I am duty bound to stress that CAMEL is currently a *prototype*. It pains me slightly to say so, because I am only too painfully aware of the dilemma that faces the would-be bibliographic revolutionary:

1. The designer needs real-world trials to discover what the world's formatting requirements actually are. This is an inductive, not a deductive process.
2. If the system is to be efficient, it must be tightly integrated. It therefore often will not be susceptible to quick fixes, and may undergo significant change in the course of its development.
3. Automated bibliography management is useless if it does not provide a complete solution to formatting problems. Until the system is complete, no one will use it for fear of holding up production or sinking work into text that relies on an unstable syntax.
4. Go to 1.

It remains to be seen whether CAMEL will break out of this circle of reticence, but before it attempts to do so it must mature further. That will require feedback. And the solicitation of feedback is one of the main purposes of this article.

That we have been able to overcome the first hurdle at all — the drafting of a working prototype of the new system — has been possible largely because of the diversity of inputs made by each of the core contributors to the bibliography team. Pedro Aphalo, the first task coordinator for bibliography support in the LaTeX3 project, was first to propose an integrated interface for handling bibliographic formats. David Rhead carried out careful research into the typesetting needs and available tools beyond the TeX world, the results of which appear in (Rhead, 1993). I joined the team after writing the LexiTeX package, described in (Bennett, Jr., 1993), to meet my own needs as a law lecturer. Prompted in large part by new possibilities opened up by Oren Patashnik's commencement of work on BibTeX 1.0, I sought to bring my then-existing code up to, or at least close to, the standard that had been set by Pedro and David. The result was been Camel; and when the draft package hit CTAN, Matt Swift provided valuable feedback on the needs of editors of critical editions and works of literary criticism.

Camel thus arose at the confluence of two distinct streams of development, one based on the identification of general "operational requirements" in advance of code drafting, the other on the persistent refashioning and honing of what, as the person responsible for it, I can safely describe as ad hoc TeX hackery. Although the code of Camel was fashioned by me, incorporating my own preferred design choices and, no doubt, infelicities of design, even this modest effort could not have been accomplished without the groundwork and feedback offered by the other members.

The next two sections of this paper provide a descriptive overview of the LaTeX and the BibTeX user interfaces in the Camel prototype. This is followed by a final section which discusses various parts of the system which are either worthy of attention because they attempt to do something new and perhaps useful, or because they may be changed significantly in future releases. In particular, readers may wish to note the rather drastic structural changes that may become possible, following discussions with Oren Patashnik about his work on BibTeX 1.0.

2 The LaTeX User Interface

In a 1993 article in this journal (Rhead, 1993), David Rhead specified the fundamental requirements for a new citation management system, and suggested ways in which LaTeX3 might address these requirements. I have entirely accepted the specification, but have adopted a rather different method of addressing it from that recommended. The core of the specification lists five reasonably common forms for handling reference lists and cross-referencing:

1. Reference by number, with the reference-list in alphabetical order of author's names;
2. Reference by number, with the reference-list in order of first citation;
3. Reference by author and date of publication, with the reference-list sorted accordingly;
4. Short-form citations in footnotes, which refer to a reference list sorted by author and date of publication; and
5. Citation in footnotes, with the first citation giving full details.

He also noted two important capabilities that are currently beyond the capacity of vanilla LaTeX, at least where documents are typeset as a unified whole:

- The user should be offered the option of dividing the reference list into categories; and
- A document should be able to have separate reference lists for each section.

This was further supplemented by Matt Swift, commenting on the current release of Camel:

- The user should be free to place the reference list before or after the place at which the citations occur.

David's suggestion in 1993 was that (a) ample provision should be made for formatting both citation tags and the reference list using commercial bibliography support software, rather than BibTeX; and (b) unorthodox citation styles with special requirements, such as in-footnote citations without a reference list, should be treated by a separate "plug-in module" to reduce the bulk of the code, and the volume of documentation.

The arguments in support of these suggestions were pragmatic; David was concerned that the burden of supporting a wide variety of styles might overwhelm the voluntary efforts of LaTeX's maintainers. Nonetheless, Camel seeks to offer a unified interface for all bibliographic styles: one set of commands, one list of options, one user's guide. And BibTeX is at the core of its functions. Since bibliographic management involves the manipulation of data by a series of tools, it seems to me that the simpler and the more well-defined the interfaces for the exchange of data between those tools can be made, the more flexible the package will ultimately be. I therefore settled on an approach that diverges from David's original suggestions.

Much work has gone into the streamlining of the LaTeX user interface for Camel. In it, all requirements are met with just eight commands:

```
\citationstyle{<style_name>}
```

```
\citationdata{bib1,bib2...}
\bibliographymanager{<manager_name>}
\citationsubject[<opts>]{<subj>}{<header>}
\source[<opts>]{<nickname>}[<pages>]
\forcefootnotes
\newinterword{<word>}{<text>}
\printbibliography{<subj>}
```

The use and effect of these commands are explained in the CAMEL manual; I will give only the briefest of descriptions here to indicate their function.

The `\citationstyle` command is required in all documents. Much like `\documentclass`, it tells the bibliography manager what package of code to use in its formatting work. Currently only one style, `law`, exists. This command is accepted only within the scope of the `document` environment, and it may be used repeatedly, to change bibliographic styles on the fly for different sections of a document.[1]

The user may specify a database either using the `\citationdata` command or, alternatively, by using the `\bibliographymanager` command. The former serves a function similar to the `\bibliography` command in standard LaTeX bibliography processing; it tells BibTeX which `*.bib` files it should search for citation entries. The `\bibliographymanager` command accepts one argument, which is the name of an external bibliography manager that is used for the citations in the document. Currently, the command will accept the names of the "main 4" identified by David Rhead (1993, p. 27): EndNote, Papyrus, ProCite and Reference Manager. When this command is in force, LaTeX will write citation keys, delimited in the way required for the designated bibliography manager, to a special file.[2] The bibliography management software must then be used to perform a simple "find and replace" operation, to replace the keys with `*.bib` entries for the designated keys. This adds another step to processing, but requires a minimum of special setup in the alien management software and can be covered by generic documentation.

The `\citationsubject` command is used to create classified bibliographies. More than one such command may be issued for each bibliography. The mandatory arguments are the nickname of the subject category, and the actual text to use as a header when the subject category is printed in the bibliography. Best practice is to place the `\citationsubject` commands at the top of the document, in the order in which you wish the subject categories to be produced in the bibliography.[3] The optional argument may mark a subject as a sub-category of the immediately preceding subject. It may also indicate whether or not page references — to instances of the citation in the current document — should be included. In the latter case, input and output extensions must be given, to allow the data produced to be processed using `\makeindex`.

In CAMEL, the new `\source` command replaces the `\cite` command in the standard LaTeX bibliography interface. In its simplest form, the command can be invoked with the nickname of a command in the BibTeX nickname of the desired citation as its sole argument. The page number of the citation can be placed in square braces immediately after the citation. Page numbers may be separated by commas, ampersands or single hyphens (spaces are ignored). Alphabetic characters may be included, but the page or section prefix is supplied by the style; the user only needs to worry about providing the numbers.

Further options may be put in square braces between the `\source` command itself and the braced nickname. These options may be used to suppress the author's name or the title of the work where appropriate,[4] to append a volume number for a multivolume work by a single author, to specify a citation subject category by its nickname, and so forth. The citation subject category must be given for the `\source` command if, at that point in the document, one or more `\citationsubject` commands are in force.[5]

The `\source` command can be given a list of keys, as in:

```
\source{item1,item2,item3}
```

In this case, leading options and trailing page numbers will be ignored, and a default syntax for joining characters is used. Alternatively, `\source` commands may be joined using a single bridging word:

```
\source{item1} but-see \source{item2}
```

A chain of `\source` commands joined in this way will expand as a single macro, and the bridging words will expand into a phrase with text, punctuation and typefaces determined by the style package in force at the time.[6]

[1] This function is not yet available in the prototype.

[2] The naming convention for this file has not yet been fixed. Currently it is always called `camel.bib`.

[3] A sample is included in a test file contained in the CAMEL distribution.

[4] These options will, of course, only take effect in a short-form citation style.

[5] The plan at the moment is to make the assignments relating to citation subjects local, so that their scope can be limited by enclosing them in an environment.

[6] This facility is available in the prototype. A facility for providing easy access to the list of bridging words is not yet in place, and needs to be installed.

The `\forcefootnotes` command causes LaTeX to push the expanded text of all `\source` commands into footnotes. This will be useful, for example, for authors who wish to be able to submit an article alternatively to a social science journal that requires an author-date reference list, or to a law journal that requires in-footnote citations with no reference list; printed output in both forms can be produced from a single source file, simply by changing the `\citationstyle` declaration — assuming, of course, that a Camel style package for each journal's style exists!

The `\newinterword` command is used to create bridging words and matching text output that are not built into the Camel distribution. Existing definitions can also be overridden using this command.

Finally, the `\printbibliography` command is used to explicitly place the bibliography at a particular location in the document. This command takes a single argument, which is either the word `all`, or the nickname of a subject category. Any categories which have been associated with an input and output extension will be read from the designated file.

The eight commands of the Camel user interface open the possibility of logical citation markup of LaTeX documents, with the reduction in time spent on wheel-reinvention and the training of oneself or one's staff that that entails. All that will be required (he said) is the development of Camel style packages for all users' requirements.

3 The BibTeX User Interface

Camel uses BibTeX to generate a reference list that is carried in memory throughout the LaTeX run.[7] BibTeX is applied to the `*.aux` file in the normal way, following the first LaTeX run; but each entry written on the `*.bbl` file is written as arguments to a special control macro. To handle certain special requirements, the path of least resistance was to redraft a `*.bst` style file for Camel from scratch. In the course of doing so, I introduced certain enhancements, some of which are essential features of the new system, while others are optional extras which may or may not survive the test of community criticism.

3.1 New Entry Types

A new entry type, `@CASE`, has been added, for use in citing reported law cases. Because the content of citations to case reporters varies widely between jurisdictions, Camel uses only this one entry type, and varies the format of the citation according to its content. A result of this approach is that there are no 'required' fields in the usual sense for `@CASE` entries — BibTeX will not complain if something is missing. Instead, there is a set of 'core' fields for each of four different formatting styles. For someone familiar with legal resources, this is actually quite intuitively straightforward.

For reported United States law cases, the core fields are `title`, `volume`, `journal`, `pages` and `year`. In addition, you may wish to specify `court`.[8] Procedural histories cannot be represented in the BibTeX entry; this must be done explicitly in the text of the LaTeX document.

For reported Commonwealth cases, use `number` instead of `volume`. The effect of this will be to place the year at the front of the citation in square braces, with the `number` and `journal` following it. Again, you may specify `court` optionally.

For cases reported in ephemeral media such as newspapers, leave out `volume` and `number`, and give the full date in the `year` field instead (see below for the formatting of dates). The formatting of the citation will adjust accordingly.

For cases reported in those jurisdictions, such as Japan, which refer to cases by date rather than by title, the citation should include `casedate`, `court`, `journal`, `volume`, `pages` and `year`. Optionally, you may also wish to include `divno` and `division`, to specify the exact identity of the deciding court.

There is a `@STATUTE` entry type, but support for statutes is still in its infancy. You need, at minimum, to enter `title`, `year` and `jurisdiction`. Jurisdictions supported in this entry type so far include `japan`, `singapore` and `england`. The present arrangement is not particularly satisfactory; and I am certainly open to suggestions on how best to manage this type of entry.

3.2 Parsing of Fields

The entry of dates has been considerably simplified in Camel BibTeX entries. Always use the `year` field (or, if appropriate, the `casedate` field). Months may be entered as numbers or as a three-character string. The following date forms are all valid:

```
year = {10 jun 1995},
year = {jun 10 1995},
year = {jun 1995},
year = {1995},
year = {10/06/95},
```

The prototype reads only the `year` field, and ignores `month`; this will be repaired in due course.

[7] The `harvard` bibliography package works in a similar way.

[8] This is not yet implemented, but can be and will be.

For @CASE entries, it is possible to enter multiple citations in a single BibTeX entry using a short form of citation in a field called cites, separating entries with an = character. The syntax of this field is rather flexible. There must be a core of one or more words, giving the name of the journal. This must be followed, at least, by a page number. The journal name must be preceded either by a volume number standing on its own, by a volume number followed by a slash and an issue number, or by a year in braces, followed by an issue number for that year. Where the year is not given in braces before the citation, it should be give in parens at the end. For the citation of mainstream case reports, these forms closely follow the citation conventions used in law journals:

```
cites = {[1995] 1 All ER 25
       = [1995] 2 WLR 125}

cites = {123 Cal.3d 237 (1995)
       = 124 S.W.2d 235 (1995)}
```

You can string together an arbitrary number of parallel citations in this way. The law style will correctly insert any page pinpointing information given to the \source command into the citation in the appropriate places.

4 The Future

This section discusses some of the features of the Camel system that are most interesting, or most in need of further attention before a proper production version is released. I will not delve into the specifics of the code — Camel is on CTAN for anyone who wants to take a look at it — but will indicate why I feel a particular portion of the code is worthy of examination, or in need of further attention. This will, I hope, give the reader a clear sense of how settled or unsettled the code is at this phase of its development.

4.1 Things unlikely to change

The most complex set of code in Camel is the expansion of the \source command. Apart from incidental concerns about the code used for parsing options (see below), I am very happy with the performance of \source and its friends. The key to its robustness is a hidden command which I have named \@ifoverword, which was developed for Camel on the basis of the \@ifnextchar command in standard LaTeX2ε. This command, like \@ifnextchar, takes three arguments: a comparison token, an argument to execute if the comparison is true, and an argument to execute if it is false. The macro absorbs all text following itself, up to a space. It then reads the following token, and returns true if it matches the comparison token, false otherwise. Within the true-text and the false-text arguments, the string that is swallowed to get at the text token used in the comparison is available in the macro \@overword.

This command is combined with a set of routines that iteratively store nickname keys and their matching arguments to list macros, so long as additional \@ifoverword matches continue to be found. These list macros are then expanded in one go, after the last match. Thus a string of source commands behaves in all respects as a single macro. Without this facility, the \forcefootnotes command would not be sufficiently robust in its effect to be of use in real-world typesetting.

Page number parsing is, I think, pretty satisfactory as written. The optional page number argument is scanned character-by-character. Each is compared with the keys in a parsing list. Those that do not match are pushed onto the end of a string, while those that do (i.e. commas, ampersands and single hyphens) are replaced with an appropriate macro before being pushed. Handled in this way, these characters behave as if they were active, but without any change to their category codes. The \source command is therefore robust, and should not break — unless some other style alters the category code of these characters.

I am also happy with the error handling routines in Camel. For a complex facility like bibliography support, detailed and specific error messages are important, and Camel does provide these where I felt help to be most clearly needed. The one qualification to this is that the error message macros do use up a large number of tokens — this while the core LaTeX3 team have been working overtime trying to reduce, one by one, the number of tokens used in the LaTeX2ε kernel. This problem will eventually be solved when the planned facility for drawing help text from external files is introduced.

For legal citations, Camel must be capable of identifying the logical context of the citation. Is this its first occurrence in the document? Is there another citation by the same author in this document? Was the immediately preceding citation to the same work? If that immediately preceding citation was in a different footnote, was it the sole citation in that footnote? And so forth. These routines have been specified with great care, and are now reasonably reliable. The intention is to embed these routines in the "kernel" of the final Camel system, so that the result of their evaluation can be drawn upon by any style that requires information concerning the context of a citation.

Finally, the structure of the Camel code has improved considerably over time. Further work is required, particularly in adding and updating commentary, but the functions performed by the package have been clearly divided into sub-units, and the internal interfaces between those units are well-defined. Further clarification of the code will be carried out, because I must involve other people in the code itself if Camel is to grow.

4.2 Possible minor changes

Options are currently specified using parsing routines that I cooked up in ignorance of the `keyval` package by David Carlisle. In due course, I plan to shift to `keyval`-based parsing. This will slightly alter the interface, by allowing mnemonic names for options to be used, instead of the single-character flags in the current prototype.

There has been some criticism of the placement of page numbers after the citation nickname in the `\source` command. I have set things up in this way because special parsing routines must be applied to page number strings. Commas, in particular, have a special significance, and would clash with the function of commas in `keyval` options unless the page number string were enclosed in braces. On balance, I feel that the current arrangement is less cluttered and easier to use. I am open to suggestions and criticism on this, however, and would welcome feedback.

4.3 A possible major change

The drafting of BibTeX 1.0, the final release, is currently being carried out by Oren Patashnik. Version 1.0 will include significant enhancements, some of which are outlined in (Patashnik, 1994). There is a possibility (at present no more than a possibility) that the finished BibTeX 1.0 will include a facility for producing "citation listings". If this facility becomes available, it will lay the groundwork for dramatic simplification and efficiency improvements in the Camel system. All that will be required (he said) is the complete rewriting of large sections of its code.

In order to cope with legal citations, Camel must alter the form of citations depending upon certain conditions, as described above in Section 4.1. The way this functions at present is for LaTeX first to pass a list of citations to the `*.aux` file, for BibTeX to generate a `*.bbl` file containing "`\lexibib`" commands and arguments including pre-parsed citation elements. The `\lexibib` commands are read in by LaTeX during the next run. Their effect is to create a set of list macros in TeX's memory, one for each citation. Each `\source` command is then expanded into the form appropriate to its logical context, using the details stored in these list macros.

This approach is wasteful of memory. A citation that occurs only once will be carried in memory throughout the second LaTeX run. For works, such as a legal textbook, that may contain thousands of citations, the cost in memory will be excessively large. And on machines with limited memory capacity, Camel will simply break on large jobs. Unfortunately, this situation cannot be avoided with current tools; BibTeX is designed to produce *reference lists*, writing one entry on output for each unique citation key. To use that information more than once — to be able to refer to a source more than once — all of the details that might be used must be stored in memory. In a short-form citation style, that means that *everything* must be stored in memory.

If a facility for producing citation lists were introduced into BibTeX, a style could instruct it to produce, in addition to ordinary pre-sorted bibliography listings, a listing of citations, one per line, marked up ready for immediate printing through LaTeX, without the need for any further parsing or post-processing. The latter would be read in one line at a time by the `\source` commands contained in the document.

This implies that (a) most of the context-identification and selective formatting work currently carried out in LaTeX would need to be recast in BibTeX code, and (b) there must be a method for passing the essential information concerning context from LaTeX to BibTeX. This is not as daft as it may at first seem; the interaction of BibTeX and LaTeX is quite difficult to grasp in standard LaTeX. It is even more difficult, if anything, in Camel. My work on the `*.bst` library for Camel suggests that the postfix stack language used by BibTeX can be tamed by the introduction of a library of high-level functions. And if all text formatting is carried out in a single forum governed by a single language, it will be much easier to see what it going on, and to work out bugs that appear in the course of the system's use. Apart from this, such an approach would have the following advantages:

- The bulk of the Camel style code in LaTeX would be substantially reduced. All of the code relating to the selection of citation elements and bridging punctuation would be eliminated.
- The marginal cost in memory and TeX tokens of each additional citation would be *zero* — less even than in the standard bibliography configuration of current LaTeX2$_\varepsilon$.

References

Bennett, Jr., Frank G. "LexITEX: context-sensitive legal citations for LaTeX". *TUGboat* **14**(3), 187–195, 1993.

Patashnik, Oren. "BibTeX 1.0". *TUGboat* **15**(3), 269–273, 1994.

Rhead, David. "The "operational requirement" for support of bibliographies". *TUGboat* **14**(4), 425–433, 1993.

◇ Frank G. Bennett, Jr.
Law Department
School of Oriental and African Studies
University of London
Thornhaugh Street
London WC1H 0XG
U.K.
`fbennett@rumple.soas.ac.uk`

Errata/Addendum

Introduction to FasTEX: A System of Keyboard Shortcuts for the Fast Keying of TEX

Filip Machi, Jerrold E. Marsden and Wendy G. McKay

- **UPDATE: Version–1 March 1996**

In our *TUGboat* article "Introduction to FasTEX: A System of Keyboard Shortcuts for the Fast Keying of TEX", Volume 16, December, 1995, some of the URL information is incorrect because of changes that were needed after the article went to press.

[Page 360]

3 The Macintosh Version of FasTEX The Macintosh version of FasTEX uses the shareware control panel TypeIt4Me, and the FasTEX shortcut files. You will need some version of system 7 to use FasTEX on the Mac.

The programs and files may be found on the following WEB sites.

Program	*Available from*
TypeIt4Me	http://www.hebel.net/~rettore/ ftp://ftp.hebel.net/users/ettore/
FasTEX files	http://cds.caltech.edu/~fastex ftp://cds.caltech.edu/pub/fastex

[Page 364]

4.1 System Requirements

To run FasTEX on UNIX you need:

Program	*Available from*
Expect/Tcl	ftp://ftp.cme.nist.gov/pub/expect
scedit	http://cds.caltech.edu/~fastex ftp://cds.caltech.edu/pub/fastex
FasTEX files	http://cds.caltech.edu/~fastex ftp://cds.caltech.edu/pub/fastex

◇ Filip Machi
Center for Extreme Ultraviolet Astrophysics
University of California at Berkeley 5030
Berkeley, CA 94720
USA
`fil@cea.berkeley.edu`

◇ Jerrold E. Marsden
Control and Dynamical Systems
California Institute of Technology 104-44
Pasadena, CA 91125
USA
`marsden@cds.caltech.edu`

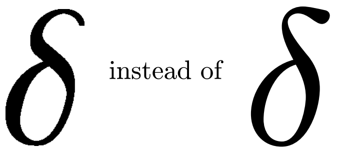
◇ Wendy G. McKay
Control and Dynamical Systems
California Institute of Technology 104-44
Pasadena, CA 91125
USA
`wgm@cds.caltech.edu`

Font Forum

Important Message Regarding CM Fonts

Donald E. Knuth[†]

If you see that your system produces the symbol



for the Greek lowercase delta, you should tell your system administrator immediately to *upgrade your obsolete version of the Computer Modern fonts*.

I made important corrections to all those fonts in the spring of 1992, but alas, I still see many books, journals, and preprints using the old versions. Please help me abolish the old forms from the typefaces of the earth.

Many characters were improved in 1992, notably the arrows, which now are darker and have larger arrowheads, so that they don't disappear so easily after xeroxing. But most of the changes are rather subtle compared to the dramatic improvement in the lowercase delta. In fact, the old delta was so ugly, I couldn't stand to write papers using that symbol; now I can't stand to read papers that still do use it.

When I released the new versions in 1992, I also installed a new and improved procedure for digitizing the letterforms at low resolutions.

The new forms of the characters take exactly the same space on the page as the old, as far as TeX is concerned. So the same line breaks and page breaks will be obtained with the 1992 fonts as with the original fonts of 1986. The only difference will be that your papers will look better and they will be more pleasant to read.

These fonts are never going to change again. The best time to get rid of the old version is now. Correct METAFONT sources are widely available in TeX archives around the world, with dozens of mirror sites; they appear in directory `/tex_archives/fonts/cm/mf`. If you use dvips for printing DVI files, it is not necessary to regenerate all the fonts with METAFONT; just delete the old ones, and dvips will make new ones as needed.

You may happen to see new publications that continue to use the old delta. If so, please tell the authors of all such papers about this Web page. Their work deserves to be typeset with more beautiful letterforms.

Incidentally, my little monograph *Axioms and Hulls* was the first publication to use the final form of the Computer Modern fonts. All details of the 1992 font refinements were documented in the fourth printing of *Computers & Typesetting, Volume E: Computer Modern Typefaces*, which came out in March 1993.

[†] This item is a slightly edited version of the "Important Message to all Users of TeX" that appears on Knuth's Web page `http://cs-faculty-www.stanford.edu/~knuth/cm.html`.

Croatian Fonts

Darko Žubrinić

The aim of this note is to inform the interested reader about the possibility of obtaining several new fonts: Croatian Glagolitic (round – sometimes called Bulgarian glagolitic, angular, Baška tablet, cursive, ligatures, the Baromić broken ligatures, calligraphic), Croatian Cyrillic, stećak ornaments and Croatian wattle patterns. They represent a considerable extension of the first version of Glagolitic fonts I created in 1992 (see [2]).

Bearing in mind that the Croatian Glagolitic script has a long history of at least 11 centuries (from the 9th to the 20th), it is not surprising that there exists a large variety of handwritten and printed versions. The author has created several major fonts of type using this script.

- The so-called “round type” font, together with the corresponding numerical values, is shown in Figure 1.

In some documents there appear additional versions of some letters, e.g., the “spider-like h” ⱒ. The tops of letters appearing in this font are aligned. The “hanging style” of the round glagolitic can be illustrated e.g., by , , as it appears on the Kiiv folia (the first folio is from Croatia, written in the region of Dubrovnik, 11th century; the remaining six are from Czechia, 10th century).

- The font of the Baška stone tablet (carved in around A.D. 1100 on the island of Krk) is shown in Figure 2.

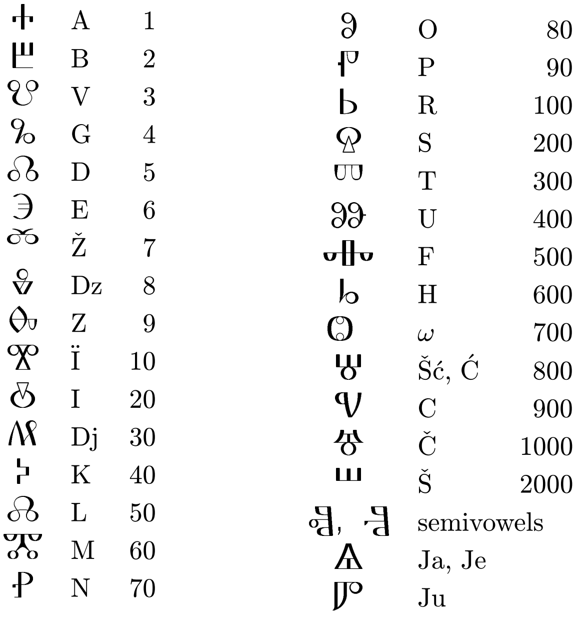

A 1
B 2
V 3
G 4
D 5
E 6
Ž 7
Dz 8
Z 9
Ï 10
I 20
Dj 30
K 40
L 50
M 60
N 70
O 80
P 90
R 100
S 200
T 300
U 400
F 500
H 600
ω 700
Šć, Ć 800
C 900
Č 1000
Š 2000
semivowels
Ja, Je
Ju

Figure 1: Croatian Glagolitic “round type”

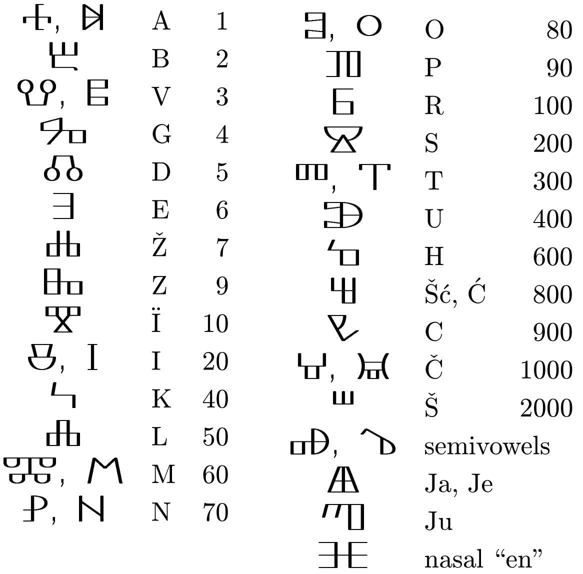
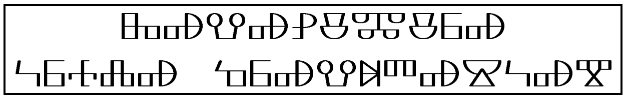

A 1
B 2
V 3
G 4
D 5
E 6
Ž 7
Z 9
Ï 10
I 20
K 40
L 50
M 60
N 70
O 80
P 90
R 100
S 200
T 300
U 400
H 600
Šć, Ć 800
C 900
Č 1000
Š 2000
semivowels
Ja, Je
Ju
nasal “en”

Figure 2: The font of the Baška stone tablet

The Baška tablet is one of our most important cultural monuments ($2\times1\ \mathrm{m}^2$). The reason is that it was written in the Croatian language (with elements of Church Slavonic) as early as the 11th century. Its text comprises more than 400 letters and contains the earliest mention of a Croatian king written in the Croatian vernacular: Z”v”nimir”, kral” hr”vat”sk”ï (Hrvatska = Croatia), i.e., *Zvonimir, the Croatian king*, or in the Glagolitic:



There are also earlier monuments from the 9th century that mention Croatian kings and dukes, but written in the Latin script and in the Latin language.

The reader will notice that several Latin and/or Cyrillic letters appear in the table: O, I, M, N, T, E (= V). This is only one among numerous proofs of the parallel use of three scripts (Glagolitic, Latin, Cyrillic) and three languages (Croatian, Church Slavonic and Latin) in Croatia. The three-script and three-language character of the Croatian Middle Ages is a unique phenomenon in the history of European culture (see [1]).

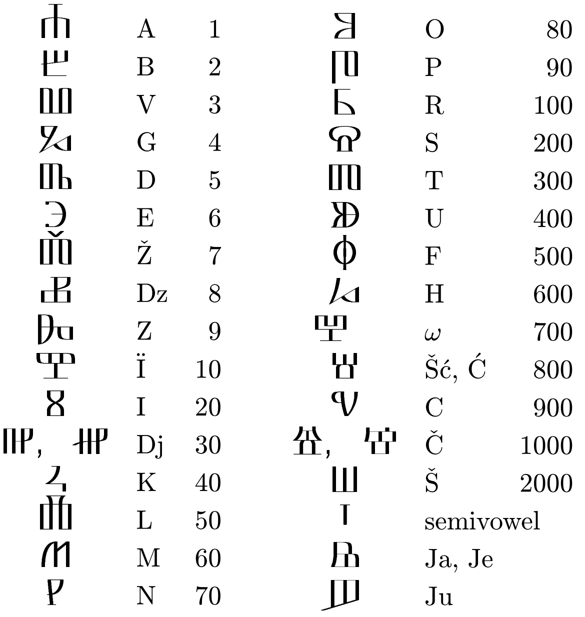

A 1
B 2
V 3
G 4
D 5
E 6
Ž 7
Dz 8
Z 9
Ï 10
I 20
Dj 30
K 40
L 50
M 60
N 70
O 80
P 90
R 100
S 200
T 300
U 400
F 500
H 600
ω 700
Šć, Ć 800
C 900
Č 1000
Š 2000
semivowel
Ja, Je
Ju

Figure 3: The angular form of Glagolitic

- Since the 12th century the Glagolitic script has survived only in Croatia. Until that time it existed also in some other regions, e.g., Bulgaria, Macedonia, Romania and Ukraine, when it was replaced by the Cyrillic script, and also in Czechia. In Croatia there developed the so-called *angular form* of Glagolitic, shown in Figure 3. Its golden period falls between the 12th and the 16th centuries. After that a decline of this script ensued, as a result of the penetration of the Ottoman Empire.

- Croatian Glagolitic has hundreds of interesting ligatures. Some of them are shown in Figure 4.
It is striking that a *printed* Glagolitic book, the Brozić breviary from 1561 (1081 pages), has as many as 250 ligatures.
- Unique in the history of European printing is the creation of the so-called *Baromić broken ligatures.*

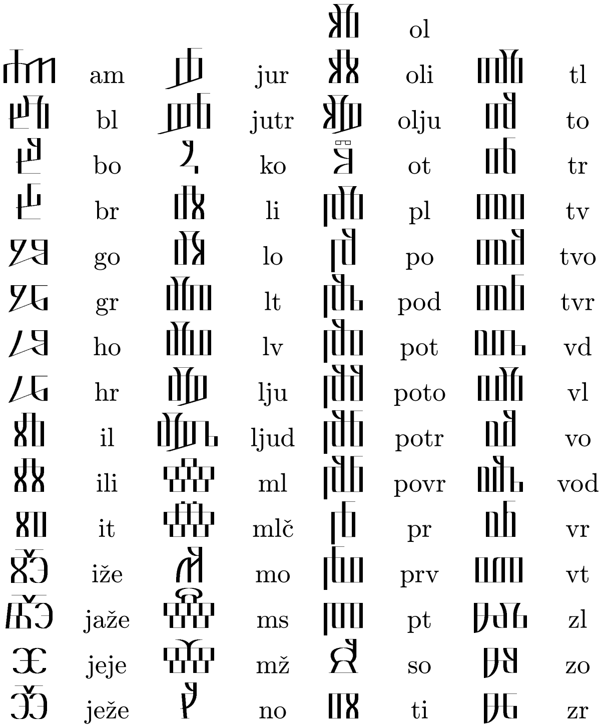

am
bl
bo
br
go
gr
ho
hr
il
ili
it
iže
jaže
jeje
ježe
jur
jutr
ko
li
lo
lt
lv
lju
ljud
ml
mlč
mo
ms
mž
no
ol
oli
olju
ot
pl
po
pod
pot
poto
potr
povr
pr
prv
pt
so
ti
tl
to
tr
tv
tvo
tvr
vd
vl
vo
vod
vr
vt
zl
zo
zr

Figure 4: Some Croatian Glagolitic ligatures

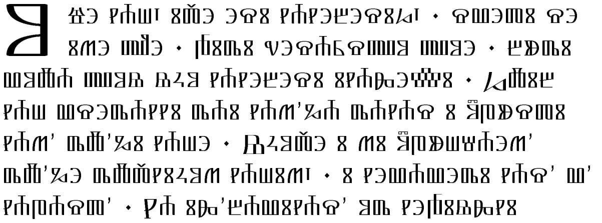
The idea was to add one half of a letter (say of) to another (say), to obtain a broken ligature (– ba). Other combinations are also possible with (, D), (, Ž), (, L), (, V), (, T). Broken ligatures were used in the Baromić Missal, the incunabulum printed in the Croatian city of Senj in 1494 (last year we celebrated its 500th anniversary). Only three samples are preserved: one in the Saltykov Ščedrin Library in St. Petersburg (Russia), one (the only complete copy) in the Szeczenyi Library in Budapest (Hungary), and one in Croatia on the island of Cres. When looking at the text containing Baromić broken ligatures, one has the impression that this incunabulum was handwritten. A few examples: (Baromić), (rain), (prayer).

Hundreds of Croatian Glagolitic texts, both handwritten and printed, the oldest from the 12th century, are held in national museums in more than 20 European countries, and also in the USA.*

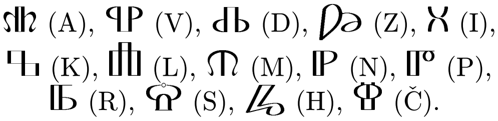
A reader living in New York can see the beautiful Croatian Missal from around 1410 in the Pierpont Morgan Library (reprinted by the Martin Sagner Verlag, Munich, 1976). In the Library of Congress in Washington you can see a sample of the first Croatian incunabulum, printed in 1483 (unfortunately, it is not known where precisely). Six copies are in Croatia, two in the Vatican Library, one in the National Library in Vienna (Austria) and one in the Saltykov Ščedrin Library in St. Petersburg (Russia). It was the first book in the history of European printing that was not printed in Latin characters, as well as the first incunabulum not printed in the Latin language.

Probably the most valuable Croatian Glagolitic book is the Missal of Hrvoje (1404), held in the Library of Turkish sultans (Topkapi Saray) in Constantinople.

A Croat, George (Juraj) de Slavonie, or de Sorbonne (14th/15th century), a professor at the Sorbonne in Paris, left us several valuable Glagolitic manuscripts written by his hand; they are held today in the Municipal Library in Reims (France). One of them contains a prayer, "Our Father", with which we would like to illustrate the Croatian Glagolitic:



• In the 16th and 17th century some calligraphic Glagolitic letters began to appear in Croatian printed books:



• There are thousands of *cursive* Glagolitic documents, bearing witness above all to the Croatian language and its very early use in official documents, and to the highly organized civil life in the Middle Ages. Probably the most important is the *Vinodol Code* from 1288. Also very important is the *Istarski razvod* from the 14th century, written in the region of Istria in three official copies: in the Latin and German languages (in Latin script), and in the Croatian language (as was expressly stated), using the Glagolitic script. It defined the borders between different rulers in Istria. Equally important are the city statutes of many Croatian cities written in the Glagolitic, the earliest dating from the 14th century. A variant of the cursive Croatian

* I would like to take the opportunity to appeal to readers: if you know of any Croatian Glagolitic documents held privately, please inform me.

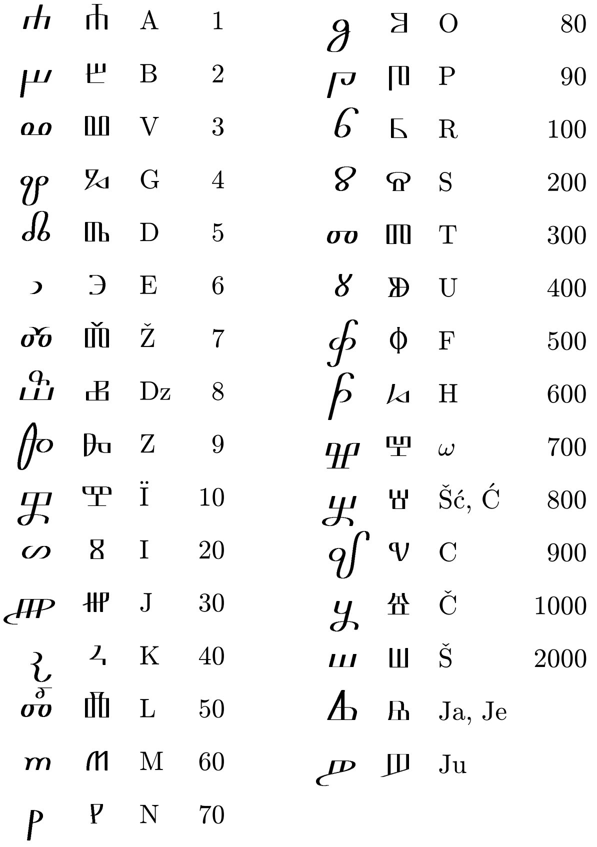
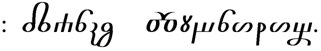
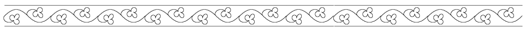

A 1
B 2
V 3
G 4
D 5
E 6
Ž 7
Dz 8
Z 9
Ï 10
I 20
J 30
K 40
L 50
M 60
N 70
O 80
P 90
R 100
S 200
T 300
U 400
F 500
H 600
ω 700
Šć, Ć 800
C 900
Č 1000
Š 2000
Ja, Je
Ju

Figure 5: A variant of cursive Croatian Glagolitic, with the angular form for comparison

Glagolitic is shown in Figure 5 (we provide also the angular type for comparison):
An example: .

• Croatian Cyrillic (also called Bosančica or Bosanica) was quite widespread among the Croats in Bosnia and in the Dalmatian part of Croatia. Its development can be traced from the 12th to 19th centuries. This script is shown in Figure 6. Probably the most beautiful Croatian Cyrillic book is the Missal of Hval, written in 1400 – 1404, now held in the University Library in Bologna (Italy), reprinted in Sarajevo in 1986.

• There exist more than 66, 000 mysterious tombstone monuments, mostly in Bosnia-Herzegovina and Croatia, called *stećak* (13th/14th century), some of them having short engravings in the Croatian Cyrillic script (e.g., “please do not disturb me, I was like you and you will be like me”), with interesting and simple border decorations, like these:



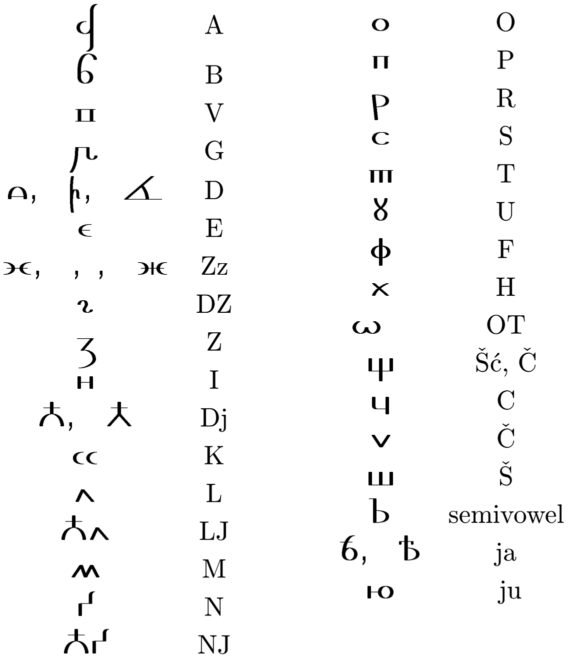
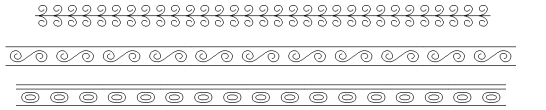
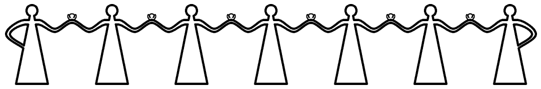

A
B
V
G
D
E
Zz
DZ
Z
I
Dj
K
L
LJ
M
N
NJ
O
P
R
S
T
U
F
H
OT
Šć, Č
C
Č
Š
semivowel
ja
ju

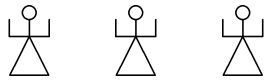
Figure 6: Croatian Cyrillic



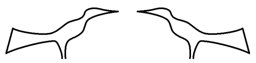
dancers:



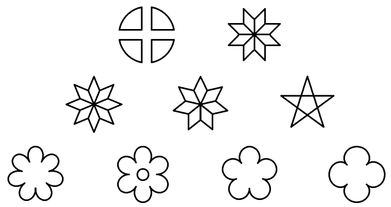
human-like figures:



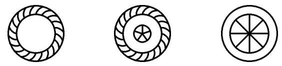
birds:



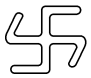
star-like ornaments:



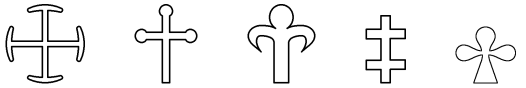
a circle (or a wheel), a symbol of eternal life:



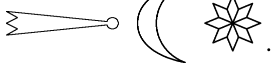
swastika:



various crosses:

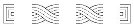


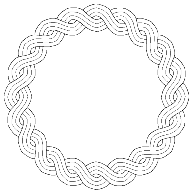
The following ornament appears frequently:

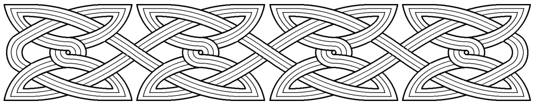


These monuments belong to Krstyans, members of the Bosnian Church, a Christian religious sect about which we still know very little.

• Let me finish this article with some of the most typical Croatian wattle patterns appearing in our pre-romanesque churches, built between the 9th and 12th centuries. From about 300 pre-romanesque Croatian churches only 15 are well preserved. The most widespread wattle pattern is this:

Of course, it is composed of . Somewhat more complex patterns are these:





Additional information about Croatian Glagolitic can be found at the URL: `http://www.tel.fer.hr/hrvatska/Croatia-HCS.html`

All the fonts appearing in this article will be available freely via the CTAN web. All ligatures and fonts are defined in the file `llig.tex` of the package. The angular Glagolitic font is activated by `\mgl` (a slightly larger version can be obtained using `\ngl`). If one wants to use Glagolitic ligatures, the definition list in `llig.tex` should be consulted. For example, to obtain ⰏⰎ (ml) one has to write `\mgl\ml`. Other fonts are activated in a similar way (they contain no Glagolitic ligatures): `\obl` (round Glagolitic; `\nobl` is a bit larger), `\bass` (Baska tablet), `\kur` (cursive), `\kal` (calligraphic), `\lom` (Baromich broken ligatures) `\hc` (Croatian Cyrillic); the several wattle patterns are activated by `\ple`, `\plet`, `\pleter`. In general, Glagolitic letters corresponding to *č, ž, š, ć (šć, št)* are encoded in TEX ligtables as `cc, zz, ss, ch`, respectively. More precisely, Ⱍ is obtained from `\mgl cc`. Capital letters are obtained by using CC (or Cc) etc. Using `cx` you can obtain yet another version of the angular č: Ⱍ. The so-called 'djerv' is obtained by typing `j` (or `J`) in the round Glagolitic. In the angular form the following two versions of the 'djerv' can be obtained: Ⰼ (`j`) and Ⰼ (`dd`). 'Yat' – Ⱑ, Ⱑ – appearing in round and angular Glagolitic can be obtained using `ya` (or `YA, Ya`). Moreover, if 'yat' is to be read as 'ye' (which is the case when it appears after a consonant), then you can use `ye` as well. The letter 'yu' Ⱓ is obtained by typing `yu`, 'izze' (Ⰺ, Ⰺ) by typing `y`. The same for the Croatian Cyrillic. Two versions of very frequently used Croatian semi-vowels can be obtained, either by typing apostrophe `'` ('yerok'; for example in ⰁⰑⰃ' 'God') or `6` ('yer'; ⰁⰑⰃⰟ). Definitions of semi-vowels 'yer' and 'yor' and other letters (like Ⰵ, ⰅⰅ) appearing in some of the oldest Croatian and Bulgarian–Macedonian documents written in the round Glagolitic can be seen in the Metafont files `oblm.mf` and `oblv.mf`, that are input in `obl.mf` — a round Glagolitic Metafont file.

As you can see, an appropriate hyphenation table is necessary to prevent line breaking between pairs of letters like `cc`, `zz`, `ss`, `ch`, `dd`, `ya`, `ye`, and it has been included with the Croatian font package too (`glhypehn.tex`). The table also incorporates some basic hypenation rules of the Croatian language. Definition names of many other symbols appearing in various fonts can be seen by looking into source files accompanying this text.

If you should use Croatian fonts, I would deeply appreciate hearing from you.

References

[1] Eduard Hercigonja: *Tropismena i trojezična kultura hrvatskoga srednjovjekovlja* [Three-Script and Three-Language Culture of the Croatian Middle Ages]. Matica hrvatska, Zagreb, 1994 (in Croatian).

[2] Darko Žubrinić: *The Exotic Croatian Glagolitic Alphabet*, *TUGboat* 13 (4), pages 470–471, 1992.

◇ Darko Žubrinić
University of Zagreb
FER, Avenija Vukovar 39
Zagreb, Croatia
Internet: `darko.zubrinic@fer.hr`

Book Reviews

Book review: Two new books on TEX in the Czech Republic

J. Veselý

With support from CSTUG, two interesting books in Czech appeared in 1995. Both authors are active CSTUG members. The author of the first book reviewed below also created the most recent CSTEX distribution, based on Eberhard Mattes' products, enhanced with a menu program that allows very user-friendly installation and daily use.

CSTUG has some difficulties with its present size (it would be helpful to have a full time employee to handle administrative work, contacts with members, publishing the journal, etc.); the last published issue of "Zpravodaj" was 4/1994. Nonetheless, CSTUG is active and interest in TEX in the Czech and Slovak Republics is growing. This is the most important thing which would stimulate efforts to overcome administrative difficulties, especially with publishing of our journal.

– – ∗ – –

Petr Olšák, *Typografický systém TEX (TEX Typesetting system)*. CSTUG, Praha, 1995, 272 pages. ISBN 80-901950-0-8.

This book is based on the author's own experiences from his work and from administration of the subnet of the Czech Technical University; it also reflects his typographical education. The book is very attractive typographically, and is bound in a soft glossy cover in Springer's yellow colour. It contains a lot of tables, pictures, and seven appendices, all prepared with great care. This is by no means a book for beginners.

The book starts with a short and simply written introduction (an ant, Fred — a popular hero of Czech fairy tales — plays a role in it), continues with a description of "different TEXs" (among other topics the macros of `plain`, AMS-TEX and LATEX are treated, as well as the role of `IniTeX` and `virTeX`), and then browses among various topics related to TEX. Special chapters are devoted to installation under DOS and UNIX, to the Czech and Slovak languages (they are similar to one another, and both are among the most heavily accented languages using the Latin alphabet), and to work with fonts (NFSS, virtual fonts, etc.). Other chapters contain information on TEX as a programming language, about METAFONT, and a few questions on the use of PostScript. The final three chapters contain explanations of how to handle pictures (METAFONT, `bm2font`, EPS), references and indexes, and how to document programs (`doc`, `C-WEB`). Much useful information is contained in the appendices, including a dictionary of TEX primitives, information about TEX archives, a list of useful books and articles, and font tables.

The coverage of the book reflects its origin: the author wrote a long series of articles on TEX for Czech Software News. Some orthodox TEXists may complain that the book is more about things related to TEX than on TEX itself. I found this fact positive: it contains material which one would search for with difficulty and which appears here in Czech for the first time. Sometimes an even more detailed explanation would be needed than the one given, but this is a book for daily use by experienced TEX-users.

– – ∗ – –

Jiří Rybička, *LATEX pro začátečníky (LATEX for beginners)*. Konvoj, Brno, 1995, ISBN 80-85615-42-8.

This book reflects the author's experience from teaching at the Technical University, Brno. It is a completely rewritten part of his lecture notes on systems for computer typesetting which the author published three years ago in collaboration with Petr Přikryl. The content of this book is in a way more standard than that of Olšák's book, but it also contains a portion of less usual subjects like treating pictures (among five appendices one is, for example, devoted to the `bm2font` program, another to a description of dvi drivers of Eberhard Mattes). It also covers the latest development of LATEX and the clear chapter structure helps in the study of the book. In comparison with the first reviewed book, this one will be really useful for beginners.

◇ J. Veselý
Charles University
Mathematical Institute
Solokovská 83
18600 Prague 8, Czech Republic
`jvesely@karlin.mff.cuni.cz`

Book review: *README.1ST*: *SGML for Writers and Editors*

Lynne A. Price

Ronald C. Turner, Timothy A. Douglass, and Audrey J. Turner, *README.1ST: SGML for Writers and Editors.* Prentice-Hall, Upper Saddle River, NJ, 1996, 241 pages + diskette, ISBN 0-13-432717-9.

The Standard Generalized Markup Language (SGML) is an increasingly popular foundation for text processing applications such as editing, formatting, archiving, and interchanging documents as well as managing document databases. While the most widely known application of SGML is HTML, the Hypertext Markup Language used to prepare documents for the Internet's World Wide Web, SGML predates the Web and is used in many other contexts as well. U.S. government agencies that use SGML include the Internal Revenue Service and the Department of Defense, while industry-wide applications of SGML exist in the aircraft, telecommunications, semiconductor, and computing communities.

The essence of SGML is that a document consists of a hierarchy of structural elements, each containing a sequence of other elements or text — a book consists of a title followed by a sequence of chapters; each chapter has a title followed by paragraphs that might be interspersed with itemized lists, figures, and tables; the paragraphs contain text, emphasized phrases, and new terms. The author using an SGML-based word processor, even a what-you-see-is-what-you-get word processor, is not required to specify the format of each element in the document; formatting is applied automatically based on the context in which the element occurs. Writing with SGML thus resembles using a well-controlled TEX macro package in which the only control sequences an author uses are those that indicate the start and end of elements. In fact, using TEX and SGML together is common. This reviewer wrote about the combination in the July 1987 issue of *TUGboat* and spoke about using TEX as a back-end in an SGML-based document production system at the TEX Users Group Annual Meeting in 1988.

SGML does not define the possible elements that can occur in a class of documents. Instead, each SGML document includes a document type definition or DTD that defines the possible elements that can occur, as well as the contexts in which each element is permitted. The DTD is followed by a document instance that contains data and markup that indicates how the document instance conforms to the DTD; the bulk of this markup consists of tags indicating the start and end of structural elements. The DTD lets software report errors if information is missing or out of place in the document instance. SGML software can also anticipate required elements to guide the user in creating document instances that conform to the DTD.

SGML is defined in International Standard ISO 8879:1986 Information Processing — Text and office systems — Standard Generalized Markup Language (SGML) and its amendment, ISO 8879:1986/A1:1988. When this standard was originally published in 1986, its terse and formal prose was the only source of information on SGML. In the years since, however, several ways to learn about SGML have evolved. There are SGML conferences and tutorials. Both public and commercial SGML software is now available. Several books describing SGML have been written. Prentice-Hall is now publishing a series of books, edited by Charles Goldfarb, editor of ISO 8879, on open information management. These books deal with managing information — particularly textual data — so that it can be processed by software beyond that planned when the data were created. SGML is a key part of this strategy. In addition to tutorials on SGML and related standards for the technical reader, the series addresses business-related topics such as justification for open information management and commerce in electronic information. The first volume in the series to appear is *README.1ST: SGML for Writers and Editors* by Ronald C. Turner, Timothy A. Douglass, and Audrey J. Turner.

In his foreword to this book, Goldfarb observes it is "easy and fun to read." This reviewer agrees. It is a fun book. It is well written and presents ideas in a logical progression. It is not, however, a textbook on SGML. Rather, it is a textbook on information management. It presents the motivation, philosophy, and history that led to the development of SGML. It describes the value of standards, the benefits of open-ended document processing, and the advantages of formal tagging. In the process, highlights of SGML are introduced where appropriate in the context of the ongoing discussion.

The book begins with background, discussing writing with standards, the SGML paradigm, and features and benefits of the language. The topic next switches to document analysis, as the authors explore the structure of a short sample document to determine the structural elements that occur within it and the contexts in which those elements occur. After a discussion of document type definitions and their use, this analysis is then used to build a DTD which the authors call WAE for "Writers and Editors". The next several chapters present major aspects of DTDs, gradually extending the WAE DTD as new concepts are introduced. The ideas of element declarations, entities, attributes, hypertext, and marked sections are gradually developed. The attribute chapter, for instance, gives several practical examples of attributes, instead of the one or two quick examples given in some other expositions of SGML.

Once this SGML foundation is established, the authors test it by walking the reader through a reading of a more complex DTD. The one they select is the publicly available DocBook DTD, intended for software documentation. The book concludes with a discussion of HTML as an SGML application and a quick introduction to hypermedia applications of SGML using another standard, the Hypermedia/Time-based Structuring Language or HyTime (ISO/IEC 10744:1992). Several appendices include the SGML markup for one chapter in the book and the DTD used to produce it. A diskette accompanying the book provides the DOS user with several sample SGML documents, a viewer for inspecting formatted versions of WAE documents, an SGML parser, the DocBook and HTML DTDs, and a list of Web sites with SGML information.

Much of the material in the book could be used to stimulate the kind of invigorating debate that sometimes occurs around a cafeteria lunch table. For example, the authors claim that the advent of desktop publishing was not a true paradigm shift, because users of desktop publishing systems think of their task as the production of pages, just as did earlier authors. They believe that use of SGML is a more fundamental change, because its users think differently about the deliverable they are producing — it is an information product that may or may not be realized on physical pages. While classifying the profundity of evolving technology is clearly subjective, the authors overlook an important characteristic of computerized text processing. Before desktop publishing, an author expected his work to be typeset by a printer — someone else had to be involved in making the work available to its audience. One of the important differences enabled by current tools is the ability of the author to produce camera-ready copy. Furthermore, the authors point out that SGML divorces a document from processes performed upon it. They neglect to point out that editing a document is a process. While observing that formatting a document for printing or screen display can be completely independent of the string of characters a writer sees while editing, they fail to admit that a tagged character file is formatted, too. They miss the opportunity to discuss what-you-see-is-what-you-get editing of SGML documents. Does reading markup affect an author's perception of his own words? They even recommend against indenting SGML markup to indicate the hierarchy of elements in a document because of possible misinterpretation of the resulting white space. Instead, they should have suggested that DTD developers consider the markup style of their users and that SGML formatting applications discard extra space and tab characters.

The book stresses the benefits of standardized DTDs. All the DTDs that appear in the book are very general. This emphasis eliminates one of the advantages of SGML. A DTD can enforce many of an organization's writing standards, requiring, for instance, that at least one introductory paragraph precede the first figure in a section, or that there be at least two sections at each level of subdivision. The WAE DTD is so flexible that it does not even require that a manual have a title. While this approach is a perfectly legitimate application of SGML, the book would have been improved by illustrating a more restrictive DTD philosophy as well. Naming style is another area of personal preference long debated in programming circles. The WAE DTD uses abbreviated names that the reader must repeatedly translate: `ti` for title, `au` for author, and so forth. Ironically, the DTD the authors used to produce the book, provided in Appendix C, includes more complete names. There is no mention of recursive element types. A subdivision of a section is a subsection (or an `ssec`) and not also a section. There are valid reasons for both approaches, and a richer presentation would have at least illustrated both strategies.

Minor improvements could be made in a handful of other areas. The authors refer several times to markup minimization, a general term that refers to several optional SGML conventions for abbreviating markup. One form of markup minimization is tag omission in which the markup that indicates the start or end of an element is omitted. The authors refer several times to this form of minimization without explaining any of the others. Furthermore, discussion of the "omitted tag minimization parameter" suggests that this construct allows tags to be omitted. It should have been made very clear that tags can only be omitted when the DTD provides for their omission and the document instance is such that an SGML parser, following a rigorously defined algorithm, can recognize the start or end of the element without the tags.

Better comments could be used throughout the sample DTDs. The WAE DTD contains only a few comments that label major syntactic components. There are no comments explaining the abbreviations used in the short names or the intended semantics of the declared elements. There are no comments at all in the DTD used to markup the sample chapter. Finally, a windows-oriented interface to the material on the diskette would make these samples more accessible to the intended nontechnical end users.

Despite these criticisms, or perhaps because the subject matter encourages the SGML-knowledgeable reader to be aware of stylistic preferences, the book is of interest to both SGML novices and to experienced practitioners. Should *README.1ST* be read first? That depends on the audience. It is intended for nontechnical end users: people who will be writing (and perhaps reading) document instances, who may read DTDs, but who will not be writing them. As the title suggests, the level of detail is directed at writers and editors. The book is not a tutorial for DTD developers, implementors of SGML applications, programmers who want to write SGML parsers, TeX macro writers, or even writers and editors who have already grasped the philosophy underlying SGML. Readers in the latter group will prefer to learn SGML from a text that presents SGML detail more directly. Once they are familiar with SGML syntax, however, they may well enjoy the perspective of *README.1ST*.

◇ Lynne A. Price
Text Structure Consulting
lprice@ix.netcom.com

Letters

There's still something missing...

I have just read [Michel Goossens'] Opening Words in *TUGboat* 16, no. 4 (December 1995). Thanks for [the] nice article.

Let me point out a few things.

I am most thankful for the LaTeX 3 group, and for the work of bringing together all variants. However, there still are other developments that you mention — Omega, ε-TeX. I don't know if I am an exception, but the fact is that it is practically impossible for me to devote much time to installing various new systems and try them, and users want one TeX and one LaTeX. So I must admit I did not install Omega or ε-TeX.

Omega is said to include a multi-language environment ... etc., but we still do not have right-to-left capabilities inherent in the official TeX, and this is essential for a right-to-left language. About ε-TeX, you say, among other things, "... bi-directional typesetting ..." Is ε-TeX the TeX for right-to-left typesetting? This is perplexing. And what is the meaning of "additional control over expansions, rescanning tokens," etc.? Should ε-TeX be used instead of TeX? And if so, how about bringing together all variants?

We are still quite "backwards" in having a LaTeX which works well in a bi-directional environment. We have an old `hebrew.sty` which only works, far from perfectly, with LaTeX 2.09. As long as right-to-left capabilities are not an integral part of TeX and LaTeX, some major part is missing for us.

Thanks and all the best,

Rama Porrat
The Hebrew University of
  Jerusalem
Jerusalem, Israel
`rama@cc.huji.ac.il`

Resources

Electronic News from the Family

Mimi Burbank and Michel Goossens

Abstract

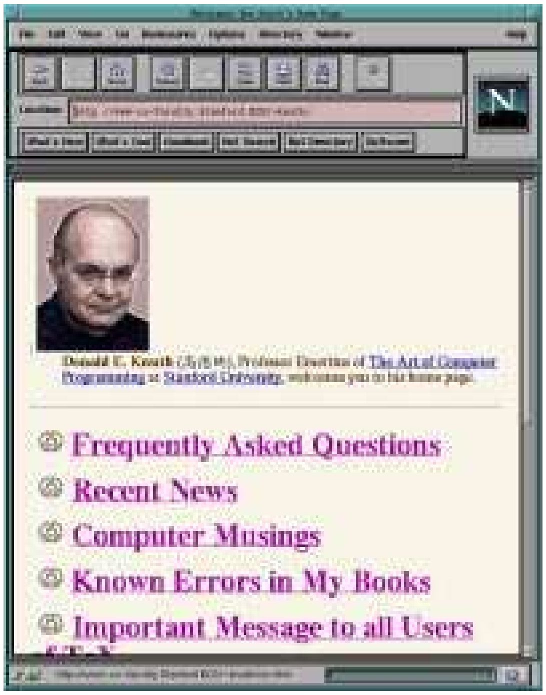
This article presents the information various TEX User Groups make available on the World Wide Web. In each case we show a reduced picture of the homepage (when available), and we present administrative information about how to contact them.

Introduction

The world of the Internet has greatly increased our ability to access information, and as TeX users are, by nature, computer users, access to information about the various User Groups around the world is one which brings the TeX community together in a unique way. These World Wide Web (WWW) pages are maintained by volunteers of the various user groups. Do you want to know more about Italian hyphenation patterns? Well, just get on the Internet and visit the web site of *iTeXnici* at `http://www.dsi.unimi.it/Users/Students/pensa/tex/uitug.html`.

The TeX Wizard himself

Donald Knuth, the father of TeX and METAFONT has his own homepage, `http://www-cs-faculty.Stanford.EDU/\char‘\~knuth/`, where you can find all kinds of interesting facts about the his life and work. (He still won't accept e-mail, and an entry in these pages tells why.)



Donald E. Knuth, Professor Emeritus of The Art of Computer Programming at Stanford University, welcomes you to his home page.

- Frequently Asked Questions
- Recent News
- Computer Musings
- Known Errors in My Books
- Important Message to all Users of TeX

Figure 1: Home page of Don Knuth

CSTUG

Československé sdružení uživatelů TeXu
(TeX users of the Czech and Slovak Republics)
Dr. Karel Horák, President
CSTUG, c/o MÚK UK, Sokolovská 83
CS-186 00 Praha 8, Czech Republic

e-mail	`horakk@earn.cvut.cz`
ftp	`ftp.muni.cz/pub/tex`
WWW	`http://www.muni.cz/TeXhelp/TeX-homepage.html`

DANTE

DANTE e.V.
Deutschsprachige Anwendervereinigung TeX e.V
(German-speaking TeX User Group)
Joachim Lammarsch, President
Postfach 101840
D-69008 Heidelberg, Germany

Tel:	+49 06221/29766
Fax:	+49 06221/167906
e-mail	`dante@dante.de`
ftp	`ftp.dante.de`
WWW	`http://www.dante.de/`

CyrTUG

Associaciia Pol'zovateleĭ Kirillicheskogo TeX'a
(Groups of Users of Cyrillic TeX)
Evgeniĭ Pankratiev, President
Irina Makhovaya, Executive Director
Mir Publishers
2, Pervyĭ Rizhskiĭ Pereulok
Moscow 129820, Russia

Tel:	095 286-0622, 286-1777
Fax:	095 288-9522
e-mail	`cyrtug@mir.msk.su`

Estonian User Group

Enn Saar, Tartu
Astrophysical Observatory, Toravere
EE 2444 Estonia
e-mail `saar@aai.ee`

GUST

Polska Grupa Użytkowników Systemu TeX
(Polish Group of TeX Users)
Tomasz Przechlewski (President)
Centrum Fizyki Teoretycznej PAN
Aleja Lotników 32/46
PL-02-668 Warszawa, Poland

e-mail	`GUST@ifpan.edu.pl`
Web	`http://www.gust.org.pl/GUST/GUSThp.html`

GUTenberg

Association GUTenberg
Groupe francophone des Utilisateurs de TeX
(Group of French-speaking TeX Users)
Michel Goossens (President)

BP 10
F-93220 Gagny principal, France

Tel: +33 1 44 32 37 96
Fax: +33 1 44 32 20 80
e-mail gut@irisa.fr

ftp ftp.univ-rennes1.fr
ftp.loria.fr
Web http://www.ens.fr/gut

GUTH (group being formed)

Grupo de Usuarios de TeX Hispanoparlantes
(User Group of Spanish-speaking TeX Users)
Julio Sanchez
GMV SA
Isaac Newton 11
PTM Tres Cantos
E-28760 Madrid, Spain

e-mail jsanchez@gmv.es
Web http://gordo.us.es/Actividades/GUTH

ITALIC (Irish)

No formal user group yet.
Public discussion list: ITALIC-L on listserver
listserv@irlearn.ucd.ie

ITeXnici (group being formed)

Italian TeX User Group
Contact: Giovanni MenoZero Pensa

e-mail pensa@dsi.unimi.it
Web http://www.dsi.unimi.it/Users/Students/pensa/tex/uitug.html

Lithuanian TeX Users Group

Lithuanian TeX Users Group
Dr. Vytas Statulevicius, Chair
Akademijos 4
LT-2600 Vilnius, Lithuania

Tel: +370 2 729 609
Fax: +370 2 729 804
e-mail vytass@ktl.mii.lt
Web http://vtex.mii.lt/tex/

NTG

Nederlandstalige TeX Gebruikersgroep
(User Group of Dutch-speaking TeX users)
Erik Frambach, Chair
Postbus 394
NL-1740 AJ Schagen, The Netherlands,

Fax: +31 26 3210500
e-mail ntg@nic.surfnet.nl
Web http://www.ntg.nl/ntg/

NTUG

Nordic TeX Users Group (Scandinavian countries)
Dag Langmyhr, Chair
Department of Informatics
P.O. Box 1080 Blindern
University of Oslo
N-0316 Oslo, Norway

Tel: +47 22 85 24 50
Fax: +47 22 85 24 01
e-mail nordictex@kth.se
Web http://www.ifi.uio.no/~dag/ntug/

TeXCeH

Slovenska skupina uporabnikov TeXa
(Slovenian TeX User Group)
Bojan Golli, President
University of Ljubljana
Jadranska 19
SI-1111 Ljubljana, Slovenia
e-mail texceh@uni-lj.si
Web http://vlado.mat.uni-lj.si/texceh/TeXCeH.HTM

UKTUG

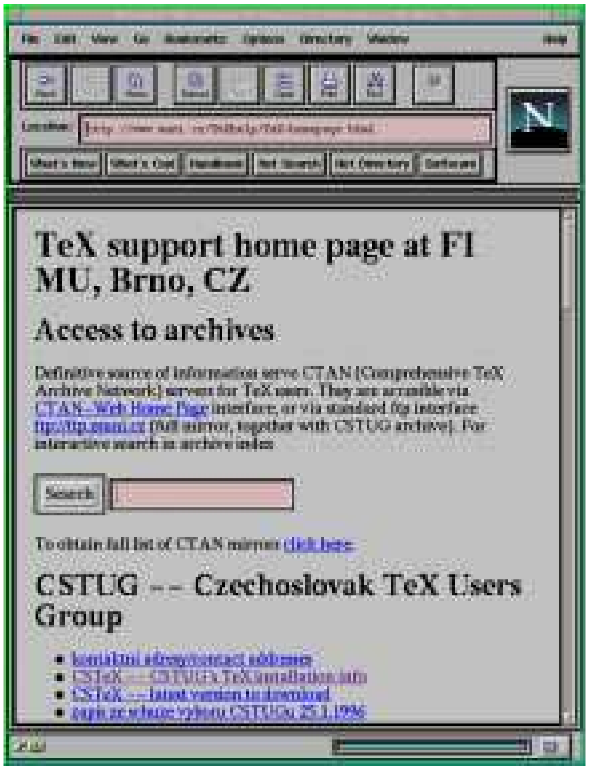
United Kingdom TeX Users Group
Robin Fairbairns, Chairman
For information:
Peter Abbott
1 Eymore Close, Selly Oak
Birmingham B29 4LB, England
Fax/Tel: +44 0121 476 2159
e-mail uktug-enquiries@tex.ac.uk
Web http://www.tex.ac.uk/UKTUG

TUG

TeX Users Group
Michel Goossens, President
Patricia Monohon, Executive Director
1850 Union St., Suite 1637
San Francisco, CA 94123 USA
Tel: +1 415 982 8449
Fax: +1 415 982 8559
e-mail tug@tug.org
ftp ftp.tug.org
Web http://www.tug.org/

◇ Mimi Burbank
SCRI, FSU, USA
mimi@scri.fsu.edu

◇ Michel Goossens
CERN, Geneva, Switzerland
goossens@cern.ch


TeX support home page at FI MU, Brno, CZ

Access to archives

Definitive source of information serve CTAN (Comprehensive TeX Archive Network) servers for TeX users. They are accessible via CTAN–Web Home Page interface, or via standard ftp interface ftp://ftp.muni.cz (full mirror, together with CSTUG archive). For interactive search in archive index

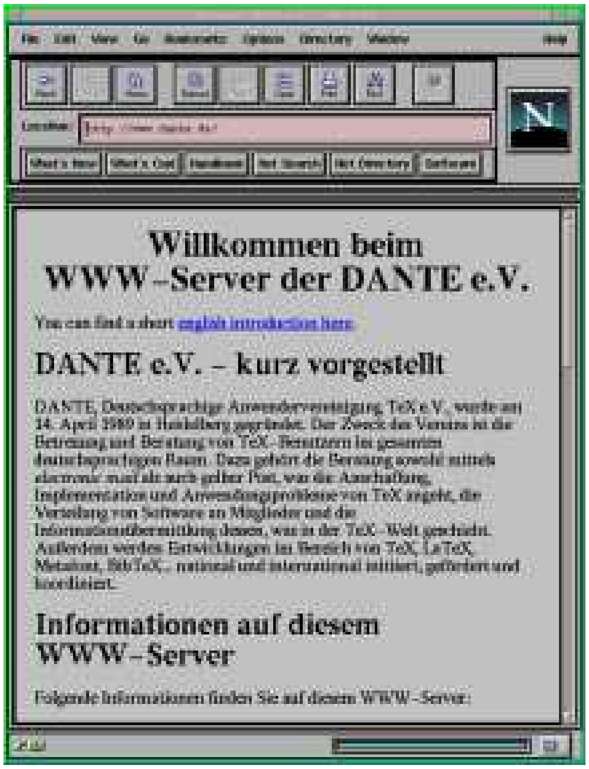
Search

To obtain full list of CTAN mirrors click here.

CSTUG -- Czechoslovak TeX Users Group

- kontaktní adresy/contact addresses
- CSTeX -- CSTUG's TeX installation info
- CSTeX -- latest version to download
- zápis ze schůze výboru CSTUGu 25.1.1996

Figure 2: Home page of $\mathcal{CS}$TUG


Willkommen beim WWW–Server der DANTE e.V.

You can find a short english introduction here.

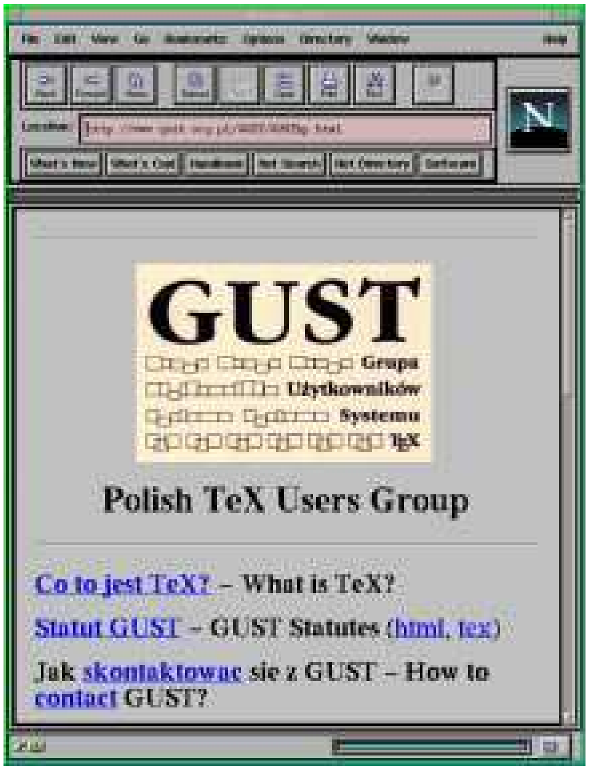
DANTE e.V. – kurz vorgestellt

DANTE, Deutschsprachige Anwendervereinigung TeX e.V., wurde am 14. April 1989 in Heidelberg gegründet. Der Zweck des Vereins ist die Betreuung und Beratung von TeX–Benutzern im gesamten deutschsprachigen Raum. Dazu gehört die Beratung sowohl mittels electronic mail als auch gelber Post, was die Anschaffung, Implementation und Anwendungsprobleme von TeX angeht, die Verteilung von Software an Mitglieder und die Informationsübermittlung dessen, was in der TeX–Welt geschieht. Außerdem werden Entwicklungen im Bereich von TeX, LaTeX, Metafont, BibTeX,.. national und international initiiert, gefördert und koordiniert.

Informationen auf diesem WWW–Server

Folgende Informationen finden Sie auf diesem WWW–Server:

Figure 3: Home page of DANTE



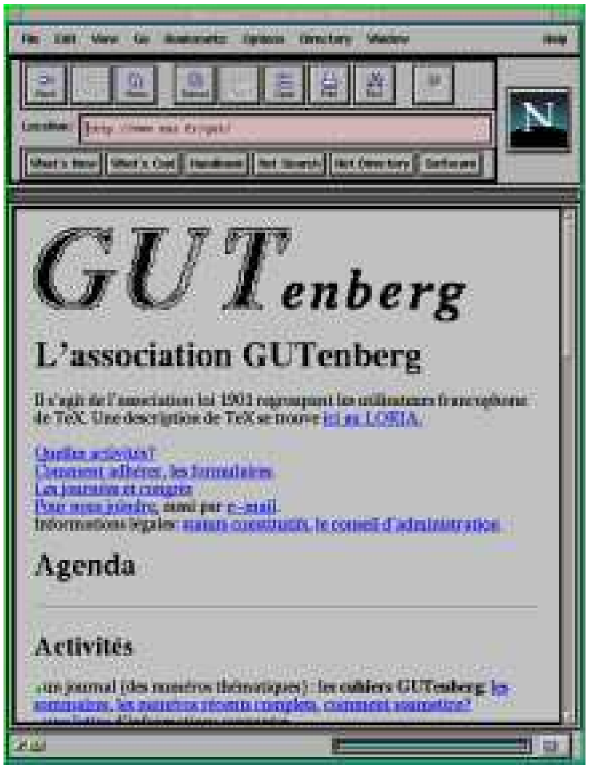
Polish TeX Users Group

Co to jest TeX? – What is TeX?

Statut GUST – GUST Statutes (html, tex)

Jak skontaktować się z GUST – How to contact GUST?

Figure 4: Home page of GUST



L'association GUTenberg

Il s'agit de l'association loi 1901 regroupant les utilisateurs francophones de TeX. Une description de TeX se trouve ici au LORIA.

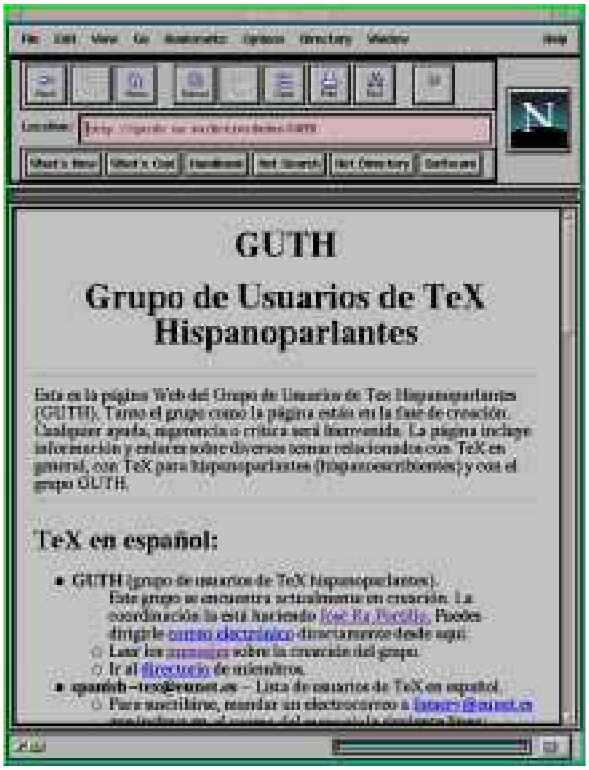
Quelles activités?
Comment adhérer, les formulaires
Les journées et congrès
Pour nous joindre, aussi par e-mail.
Informations légales: statuts constitutifs, le conseil d'administration.

Agenda

Activités

un journal (des numéros thématiques) : les cahiers GUTenberg: les sommaires, les numéros récents complets, comment s'abonner?

Figure 5: Home page of GUTenberg


GUTH

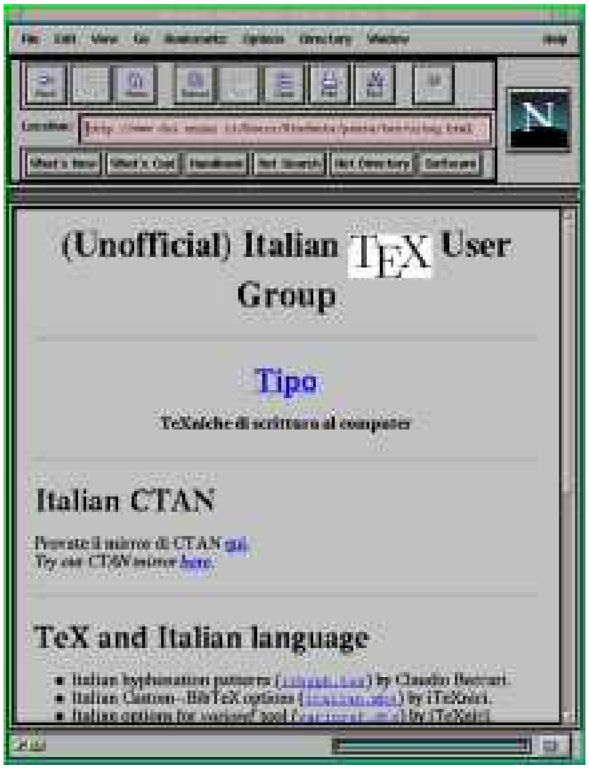
Grupo de Usuarios de TeX Hispanoparlantes

Esta es la página Web del Grupo de Usuarios de Tex Hispanoparlantes (GUTH). Tanto el grupo como la página están en la fase de creación. Cualquier ayuda, sugerencia o crítica será bienvenida. La página incluye información y enlaces sobre diversos temas relacionados con TeX en general, con TeX para hispanoparlantes (hispanoescribientes) y con el grupo GUTH.

TeX en español:

- GUTH (grupo de usuarios de TeX hispanoparlantes).
  Este grupo se encuentra actualmente en creación. La coordinación la está haciendo José Ra Portillo. Puedes dirigirle correo electrónico directamente desde aquí.
  ○ Leer los mensajes sobre la creación del grupo.
  ○ Ir al directorio de miembros.
- spanish-tex@eunet.es – Lista de usuarios de TeX en español.
  ○ Para suscribirse, mandar un electrocorreo a listserv@eunet.es

Figure 6: Home page of GUTH


(Unofficial) Italian TEX User Group

Tipo

TeXniche di scrittura al computer

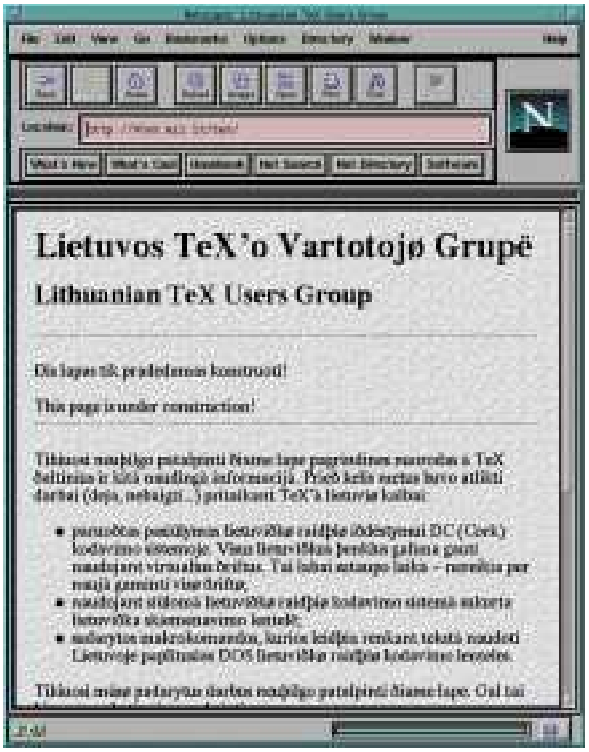
Italian CTAN

Provate il mirror di CTAN qui.
Try our CTAN mirror here.

TeX and Italian language

- Italian hyphenation patterns ([illegible]) by Claudio Beccari.
- Italian Custom–BibTeX options ([illegible]) by ITeXnici.

Figure 7: Home page of ITEXnici


Lietuvos TeX'o Vartotojø Grupë

Lithuanian TeX Users Group

Dis lapas tik pradedamas konstruoti!

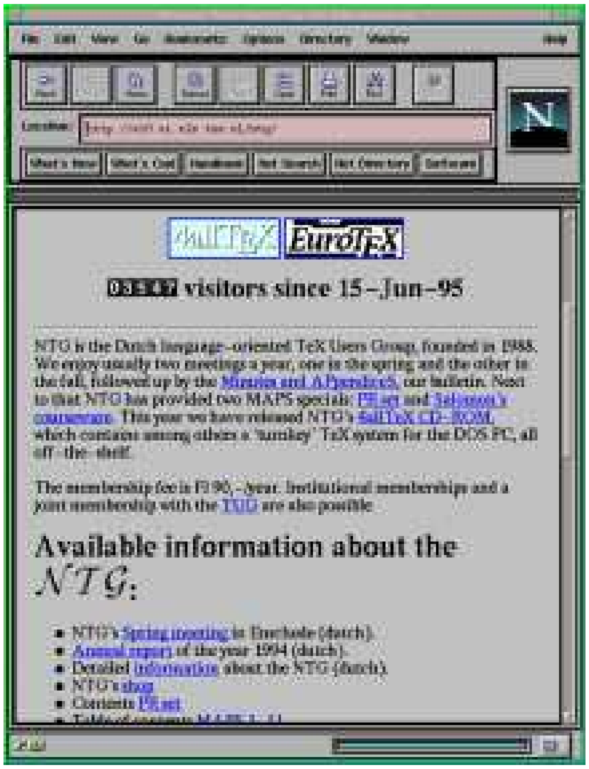
This page is under construction!

Tikimės netrukus patalpinti šiame lape pagrindines nuorodas į TeX šaltinius ir kitą naudingą informaciją. Prieš kelis metus buvo atlikti darbai (deja, nebaigti...) pritaikant TeX'ą lietuvių kalbai:

- [illegible] lietuviškų raidžių [illegible] DC (Cork) kodavimo sistemoje. [illegible]
- naudojant [illegible] lietuviškų raidžių kodavimo sistemą [illegible]
- [illegible] makrokomandos, [illegible] Lietuvoje [illegible] DOS lietuviškų raidžių kodavimo lentelės.

Tikimės [illegible] darbus netrukus patalpinti šiame lape. Gal tai

Figure 8: Home page of Lithuanian TUG



03347 visitors since 15–Jun–95

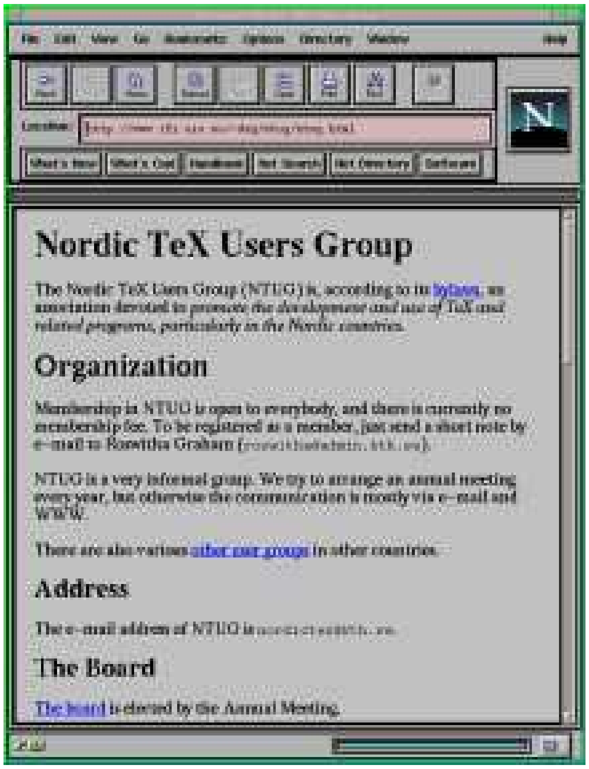
NTG is the Dutch language–oriented TeX Users Group, founded in 1988. We enjoy usually two meetings a year, one in the spring and the other in the fall, followed up by the Minutes and APpendices, our bulletin. Next to that NTG has provided two MAPS specials: PPLet and Salomon's comments. This year we have released NTG's 4allTeX CD–ROM, which contains among others a 'turnkey' TeX system for the DOS PC, all off–the–shelf.

The membership fee is Fl 90,–/year. Institutional memberships and a joint membership with the TUG are also possible.

Available information about the NTG:

- NTG's Spring meeting in Enschede (dutch).
- Annual report of the year 1994 (dutch).
- Detailed information about the NTG (dutch).
- NTG's shop
- Contents PPLet

Figure 9: Home page of NTG



Nordic TeX Users Group

The Nordic TeX Users Group (NTUG) is, according to its bylaws, an association devoted to *promote the development and use of TeX and related programs, particularly in the Nordic countries.*

Organization

Membership in NTUG is open to everybody, and there is currently no membership fee. To be registered as a member, just send a short note by e-mail to Roswitha Graham (roswitha@admin.kth.se).

NTUG is a very informal group. We try to arrange an annual meeting every year, but otherwise the communication is mostly via e-mail and WWW.

There are also various other user groups in other countries.

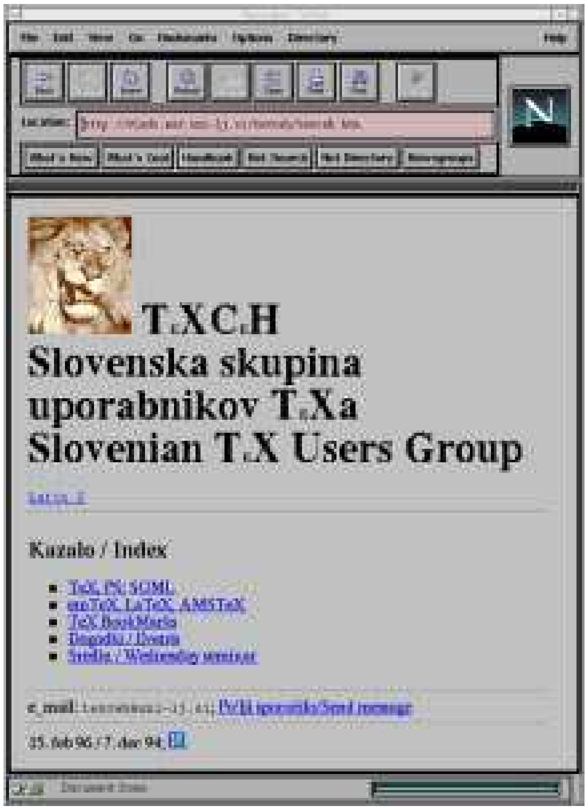
Address

The e-mail address of NTUG is [illegible]

The Board

The board is elected by the Annual Meeting.

Figure 10: Home page of Nordic TUG



TEXCEH Slovenska skupina uporabnikov TEXa Slovenian TEX Users Group

Latin 2

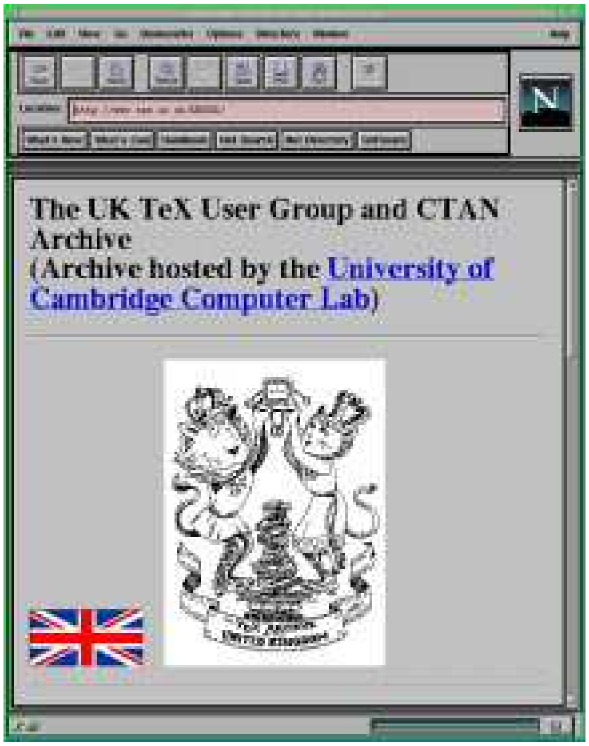
Kazalo / Index

- TeX, PS, SGML
- emTeX, LaTeX, AMSTeX
- TeX BookMarks
- Dogodki / Events
- Sreda / Wednesday seminar

e_mail: texceh@uni-lj.si; Pošlji sporočilo/Send message

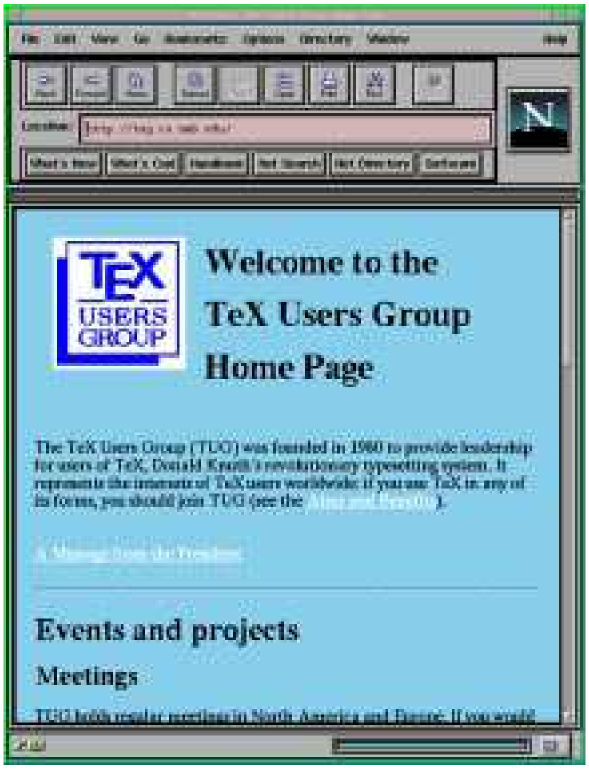
25.feb 96 / 7.dec 94;

Figure 11: Home page of TeXCeH



The UK TeX User Group and CTAN Archive (Archive hosted by the University of Cambridge Computer Lab)

Figure 12: Home page of UKTUG



Welcome to the TeX Users Group Home Page

The TeX Users Group (TUG) was founded in 1980 to provide leadership for users of TeX, Donald Knuth's revolutionary typesetting system. It represents the interests of TeX users worldwide: if you use TeX in any of its forms, you should join TUG (see the Aims and Benefits).

A Message from the President

Events and projects

Meetings

TUG holds regular meetings in North America and Europe. If you would

Figure 13: Home page of TUG

Tutorials

Using EPS Graphics in LaTeX 2$_\varepsilon$ Documents Part 1: The graphics and graphicx packages

Keith Reckdahl

Abstract

This is the first of two papers that explain how to use Encapsulated PostScript (EPS) files in LaTeX 2$_\varepsilon$ documents.

The graphics and graphicx packages provide commands which insert, scale, and rotate EPS graphics. In addition to graphic-insertion commands, commands which are commonly used in conjunction with EPS graphics are covered.

Compressed EPS files and non-EPS graphic formats (TIFF, GIF, JPEG, PICT, etc.) can also be inserted when `dvips` is used.

Since neither LaTeX nor `dvips` has any built-in decompression or graphics-conversion capabilities, that software must be provided by the user.

The second paper in the series, to appear later this year, will cover

- floating figures in various configurations (including more than one figure in a single float), and the use of the subfigure package,
- creation of boxed figures, by use of the `\fbox` command, or of the facilities of the fancybox package,
- manipulation of the caption of a figure, including use of the caption2 package, and
- modifying the text within an EPS file by using the PSfrag system, for example to include mathematical symbols or equations.

1 Introduction

Inserting Encapsulated PostScript (EPS) graphics in LaTeX originally required the low-level `\special` command. To make graphic-insertion easier and more portable, two higher-level packages epsf and psfig were written for LaTeX 2.09. In epsf, the graphics insertion was done by the `\epsfbox` command, while three other commands controlled graphic scaling. In psfig, the `\psfig` command not only inserted graphics, it also scaled and rotated them. While the `\psfig` syntax was popular, its code was not as robust as `\epsfbox`. The epsfig package was created as a hybrid of the two graphics packages, with its `\epsfig` command using the `\psfig` syntax and much of the more-robust `\epsfbox` code. Unfortunately, `\epsfig` still used some of the less-robust `\psfig` code.

The epsfig package was updated to LaTeX 2$_\varepsilon$ as a stop-gap measure while the LaTeX3 team addressed the general problem of inserting graphics in LaTeX 2$_\varepsilon$. The resulting "graphics bundle" was totally re-written, and its commands are more efficient and more robust.

The graphics bundle contains the "standard" graphics package and the "extended" graphicx package. Both packages contain an `\includegraphics` command which includes graphics, but they contain *different* versions of `\includegraphics`. The syntax of the graphicx `\includegraphics` is modeled after `\psfig`, while the syntax of the graphics `\includegraphics` is modeled after the `\epsfbox` command. As a result, the `\includegraphics` command in graphicx supports scaling and rotating, but that in the graphics must be nested inside `\scalebox` and/or `\rotatebox` commands for scaling and/or rotating.

This paper has been typeset using the graphicx package because its syntax is more convenient than the graphics syntax. Since both packages have the same capabilities, the examples in this document can also be performed with the graphics package, although the resulting syntax may be more cumbersome. The syntax of the graphicx commands is described in section 5. The syntax of the graphics commands is described in section 6. For a full specification of the packages, see David Carlisle's graphics bundle documentation [1].

For backward compatibility, the graphics bundle also includes the epsfig package which replaces the original LaTeX 2$_\varepsilon$ epsfig package. The new epsfig package defines the `\epsfbox`, `\psfig`, and `\epsfig` commands as wrappers which translate to a simple call to the `\includegraphics` command.

2 LaTeX Terminology

A *box* is any LaTeX object (characters, graphics, etc.) that is treated as a unit (see [4, page 103]). Each box has a *reference point* on its left side, and a *baseline*, which is a horizontal line passing through the reference point (see Figure 1). When LaTeX forms lines of text, characters are placed left-to-right with their reference points aligned on a horizontal line called the *current baseline*, aligning the characters' baselines with the current baseline. LaTeX follows the same process for typesetting graphics or other objects; the reference point of each object is placed on the current baseline.

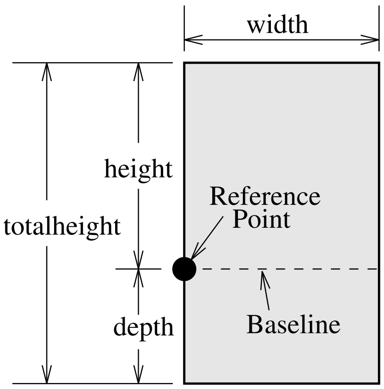


Figure 1: Sample LaTeX Box

The size of each box is described by the three lengths *height*, *depth*, *width*. The *height* is the distance from the reference point to the top of the box. The *depth* is the distance from the reference point to the bottom of the box. The *width* is the width of the box. The *totalheight* is defined as the distance from the bottom of the box to the top of the box, or totalheight = height + depth.

The reference point of a non-rotated EPS graphic is its lower-left corner (see left box in Figure 2), giving it zero depth and making its totalheight equal its height. The middle box in Figure 2 shows a rotated graphic where the height is not equal to the totalheight. The right box in Figure 2 shows a rotated graphic where the height is zero.

3 The EPS BoundingBox

In addition to PostScript graphics language commands which draw the graphics, EPS files contain a BoundingBox line which specifies the natural size of the graphics. By convention, the first line of a PostScript file specifies the type of PostScript and is then followed by a series of comments called the *header* or *preamble*. (Like LaTeX, PostScript's comment character is %). One of these comments specifies the BoundingBox. The BoundingBox line contains four numbers:

1. The x-coordinate of the lower-left corner of the BoundingBox.
2. The y-coordinate of the lower-left corner of the BoundingBox.
3. The x-coordinate of the upper-right corner of the BoundingBox.
4. The y-coordinate of the upper-right corner of the BoundingBox.

For example, here are the first 5 lines of an EPS file created by gnuplot:

```
%!PS-Adobe-2.0 EPSF-2.0
%%Creator: gnuplot
%%DocumentFonts: Times-Roman
%%BoundingBox: 50 50 410 302
%%EndComments
```

Thus the gnuplot EPS graphic has a lower-left corner with coordinates $(50, 50)$ and an upper-right corner with coordinates $(410, 302)$. The BoundingBox parameters have units of PostScript points which are $^1/_{72}$ of an inch, making the above graphic's natural width 5 inches and its natural height 3.5 inches.

Note that a PostScript point is is slightly larger than a TeX point which is $^1/_{72.27}$ of an inch. In TeX and LaTeX, PostScript points are called "big points" and abbreviated `bp` while TeX points are called "points" and abbreviated `pt`.

3.1 Converting PS files to EPS

While most PostScript files (without BoundingBox information) can be converted to EPS, there are restrictions on the PostScript commands which can be used in EPS files. For example, EPS files cannot include the `setpagedevice`, `letter`, or `a4` PostScript operators. Single-page PostScript files without any such offending commands can be converted to EPS by one of the following methods:

1. The best option is to use a utility such as ghostscript's `ps2epsi` which will read the PostScript file, calculate the BoundingBox parameters, and create an EPS file (complete with a BoundingBox) which contains the PostScript graphics. Unfortunately, ghostscript is a large package which is not trivial to install.

2. Alternatively, the BoundingBox parameters can be calculated and then either entered in the `bb` option of `\includegraphics` or a text editor can be used to insert them directly in the PostScript file's BoundingBox line. There are several ways to calculate the BoundingBox:

   (a) The `bbfig` script uses a PostScript printer to calculate the BoundingBox. `bbfig` adds some PostScript commands to the beginning of the PostScript file and sends it to the printer. At the printer, the added PostScript commands calculate the BoundingBox of the original PostScript file, printing the BoundingBox coordinates superimposed on the PostScript graphic.

   (b) Use ghostview to display the PostScript graphic. As you move the pointer around

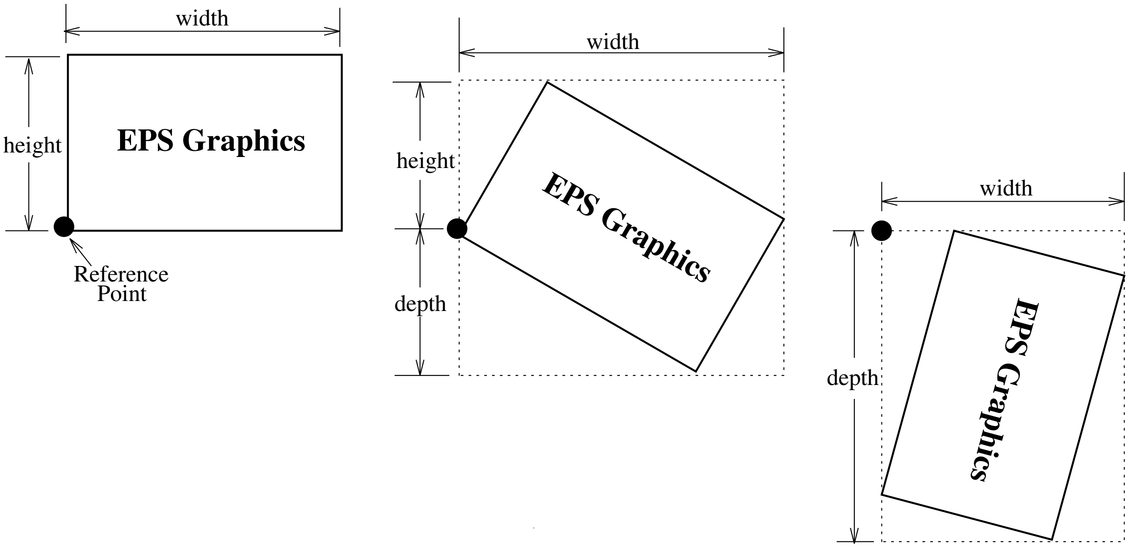


Figure 2: Rotated LaTeX Boxes

the graphic, ghostview displays its coordinates (with respect the the lower-left corner of the page). To determine the BoundingBox parameters, record the pointer coordinates at the lower-left corner of the graphic and the upper-right corner of the graphic.

(c) Print out a copy of the PostScript graphics and measure the horizontal and vertical distances (in inches) from the lower-left corner of the paper to the lower-left corner of the graphics. Multiply these measurements by 72 to get the coordinates of the BoundingBox's lower-left corner. Likewise, measure the distances from the lower-left corner of the paper to the upper-right corner of the graphics to get the coordinates of the BoundingBox's upper-right corner.

4 Graphics in DVI Files

When LaTeX documents are compiled, the graphics-inclusion commands do not insert the EPS graphics file into the DVI file. Rather, they do two things:

1. They reserve the proper amount of space for the graphic in the LaTeX document.
2. They place a file-specification command in the DVI file which specifies the name of the EPS file.

When a DVI-to-PS converter (such as `dvips`) converts the DVI file to PostScript, the file-specification command causes the converter to insert the EPS graphics into the PostScript file. Therefore,

- the EPS graphics do not appear in most DVI-viewers. To help the user with placement of the graphics, most DVI viewers display the BoundingBox in which the graphics will be inserted.
- the EPS files must be present when the DVI file is converted to PS. Thus the EPS files must accompany DVI files whenever they are moved.

5 The Commands in the graphicx Package

The coverage of the graphicx package is sporadic: [3] covers both the graphics and graphicx packages, [4] only covers the graphics package and [2] describes neither. The best reference for the graphics and graphicx packages is [1].

The graphicx package has five main commands:

```
\includegraphics[options]{filename}
\rotatebox{angle}{argument}
\scalebox{h-scale}[v-scale]{argument}
\resizebox{width}{height}{argument}
\resizebox*{width}{totalheight}{argument}
```

5.1 The includegraphics Command

Syntax:

```
\includegraphics[options]{filename}
```

Table 1 lists the geometric options available when using the `\includegraphics` command. The BoundingBox can alternatively be specified with the options `natheight` and `natwidth`. `natheight=h` with `natwidth=w` are equivalent to `bb=0 0 h w`. For backward compatibility, the BoundingBox coordinates can also be individually specified with `bbllx`, `bblly`, `bburx`, `bbury` options.

Table 1: `\includegraphics` Options

`height`	The height of the graphics (in any of the accepted TEX units).
`totalheight`	The totalheight of the graphics (in any of the accepted TEX units). *(Added 6/95)*
`width`	The width of the graphics (in any of the accepted TEX units).
`scale`	Scale factor for the graphic. Specifying `scale=2` makes the graphic twice as large as its natural size.
`angle`	Specifies the angle of rotation (in degrees) with a counter-clockwise (anti-clockwise) rotation being positive.
`origin`	The `origin` command specifies what point to use for the rotation origin. (By default, the object is rotated about its reference point.) *(Added 12/95)* The possible origin points are the same as those for the `\rotatebox` command in section 5.4. For example, `origin=c` rotates the graphic about its center.
`bb`	Specifies BoundingBox parameters. For example `bb=10 20 100 200` specifies that the BoundingBox has its lower-left corner at (10,20) and its upper-right corner at (100,200).

Table 2: `\includegraphics` Cropping Options

`viewport`	Specify what portion of the graphic to view. Like a BoundingBox, the area is specified by four numbers which are the coordinates of the lower-left corner and upper-right corner. The coordinates are relative to lower-left corner of the BoundingBox. *(Added 6/95)* For example, `viewport=0 0 72 72` displays the 1-inch square from the lower left of the graphic. Note that some early graphicx versions may have a broken `viewport` option in which `viewport=a b c d` produces an upper-right corner of (a+c,b+d) instead of (c,d).
`trim`	An alternate method for specifying what portion of the graphic to view. The four numbers specify the amount to remove from the left, bottom, right, and top side, respectively. Positive numbers trim from a side, negative numbers add to a side. *(Added 6/95)* For example, `trim=1 2 3 4` trims the graphic by 1 `bp` on the left, 2 `bp` on the bottom, 3 `bp` on the right, 4 `bp` on the top.

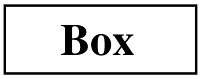
Table 3: `\includegraphics` Boolean Options

`clip`	When `clip=true` or `clip` is specified, any graphics outside of the viewing area are clipped and do not appear.
`keepaspectratio`	When `keepaspectratio=true` or `keepaspectratio` is specified, specifying both the `width` and `height` or `totalheight` options does *not* distort the graphic. Instead, the graphic is made as large as possible such that its aspect ratio remains the same and the graphic does not exceed either the specified height or width. *(Added 9/95)*
`draft`	When `draft=true` or `draft` is specified, the graphic's BoundingBox and filename is displayed in place of the graphic, saving time. See section 7.

Since `\includegraphics` automatically reads the BoundingBox parameters from the EPS file, these options are usually not specified. They are useful if the BoundingBox parameters in the EPS file are missing or are incorrect. While the `bb` option can also be used for cropping the EPS graphics, the `viewport` or `trim` options (see Table 2) are recommended. Table 3 lists other control options.

Example:

The EPS file `box.eps` contains:



The commands:

```
\documentclass{article}
\usepackage{graphicx}
\begin{document}
   Some text.
   \includegraphics{box.eps}
   More text.
\end{document}
```

produce:

Some text. More text.

Since `\includegraphics` does not end the current paragraph, it can place EPS graphics within text, for example ⊗ or ◕. The placement of the graphic is controlled by the current text justification. To center the graphic, put it inside a center environment:

```
\begin{center}
   \includegraphics[width=2in]{box.eps}
\end{center}
```

Alternately, if the `\includegraphics` command is inside an environment (such as `minipage` or `figure`), the `\centering` declaration centers the remaining output of the environment. For example:

```
\begin{figure}
   \centering
   \includegraphics[width=2in]{box.eps}
\end{figure}
```

is similar to:

```
\begin{figure}
\begin{center}
   \includegraphics[width=2in]{box.eps}
\end{center}
\end{figure}
```

The difference between these examples is that the `center` environment produces extra vertical space above and below the environment, while `\centering` produces no extra space.

5.2 The scalebox Command

Syntax:

```
\scalebox{h-scale}[v-scale]{argument}
```

The `\scalebox` command scales an object, making its width be its original width multiplied by `h-scale`. The object can be any LaTeX object: letter, paragraph, EPS graphic, etc. The object's height is its original height multiplied by `v-scale`. Negative values reflect the object. If `v-scale` is omitted, it defaults to h-scale, which keeps the aspect ratio constant.

5.3 The resizebox Commands

Syntax:

```
\resizebox{width}{height}{argument}
\resizebox*{width}{totalheight}{argument}
```

The `\resizebox` command resizes an object to a specified size. The object can be any LaTeX object: letter, paragraph, EPS graphic, etc. Specifying `!` as either height or width makes that length be such that the aspect ratio remains constant. The standard LaTeX 2$_\varepsilon$ arguments `\height`, `\depth`, `\width`, `\totalheight` can be used to refer to the original size of `argument`. So `\resizebox{2in}{\height}%` `{argument}` makes `argument` keep its same height but have a width of 2 inches.

The `\resizebox*` command only differs from `\resizebox` in its second argument, which specifies the `totalheight` of the object.

5.4 The rotatebox Command

Syntax:

```
\rotatebox[options]{angle}{argument}
```

The `\rotatebox` command rotates an object by an angle given in degrees, with a counter-clockwise rotation being positive. The object can be any LaTeX object: letter, paragraph, EPS graphic, etc. By default, the object is rotated about its reference point. The options allow the user to specify the point of rotation:

1. Specifying the `[x=xdim,y=ydim]`, the object is rotated about the point whose coordinates relative to the reference point are (`xdim,ydim`).
2. The `origin` option specifies one of 12 special points shown in in Figure 3.

   The horizontal position of the `origin` points is specified by one of three letters: `lcr` (which stand for left, center, right), while the vertical position is specified by one of four letters:

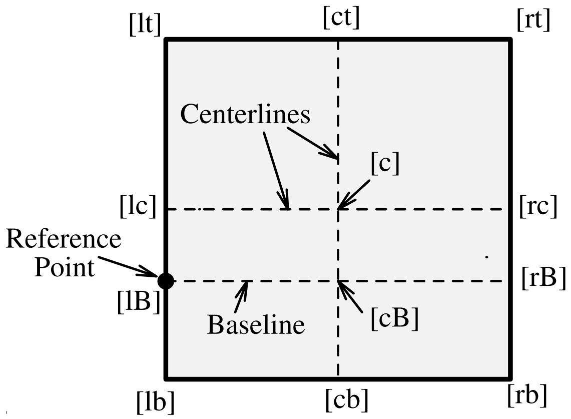


Figure 3: Available Origin Points

t,c,B,b (which stand for top, center, Baseline, bottom). For example:

[rb] specifies the bottom-right corner

[lt] specifies the top-left corner

[cB] specifies the center of the graphic's Baseline

[lc] specifies the midpoint of the left side

[ct] specifies the midpoint of the top side

Note that:

- The order of the letters is not important, making [br] equivalent to [rb].
- c represents either the horizontal center or vertical center depending what letter is used with it.
- If only one letter is specified, the other is assumed to be c, making [c] equivalent to [cc], [l] equivalent to [lc], [t] equivalent to [tc], etc.

6 The graphics Version of includegraphics

The graphics package contains two commands \includegraphics and \includegraphics* which are identical except that \includegraphics* clips (does not show) graphics outside the BoundingBox. The syntax for \includegraphics is:

```
\includegraphics[llx,lly][urx,ury]{filename}
```

[llx,lly] are the x and y coordinates of the lower-left corner of the image. [urx,ury] are the x and y coordinates of the upper-right corner of the image. If no coordinates are given, the BoundingBox in the file is used. If only one set of coordinates is given, it is assumed to be [urx,ury], with [llx,lly] set to zero. The default units for the coordinates are bp, although any valid TEX units can be used.

The graphics package's \rotatebox, \scalebox, \resizebox commands are the same as the corresponding graphicx commands except that the graphics version of \rotatebox does not allow any of the options which the graphicx version offers (see section 5.4).

The following commands use the graphicx version of \includegraphics:

```
\documentclass{article}
\usepackage{graphicx}
\begin{document}
%% include file1.eps with a width of 3 inches
\includegraphics[width=3in]{file1.eps}

%% include file2.eps with a width of 3 inches,
%%    then rotate 45 degrees
\includegraphics[width=3in,angle=45]{file2.eps}

%% include file3.eps, rotate 45 degrees,
%%    and resize to a width of 3 inches
\includegraphics[angle=45,width=3in]{file3.eps}
\end{document}
```

The following commands use the graphics version of \includegraphics to produce the same output:

```
\documentclass{article}
\usepackage{graphics}
\begin{document}
%% include file1.eps with a width of 3 inches
\resizebox{3in}{!}{\includegraphics{file1.eps}}

%% include file2.eps with a width of 3 inches,
%%    then rotate 45 degrees
\rotatebox{45}{\resizebox{3in}{!}%
                {\includegraphics{file2.eps}}}

%% include file3.eps, rotate 45 degrees,
%%    and resize to a width of 3 inches
\resizebox{3in}{!}{\rotatebox{45}%
                {\includegraphics{file3.eps}}}
\end{document}
```

7 Draft Mode in graphicx

Since LATEX documents containing PostScript figures take longer to display and print, it often is desirable to omit the actual graphic when preliminary versions of the document are viewed or printed. If the graphics or graphicx packages are used with the draft option:

```
\usepackage[draft]{graphicx}
```

then only the BoundingBox and name of any subsequent EPS graphics are displayed:

box.eps

The graphicx version of `\includegraphics` has a `draft` option which allows the user to also control this feature for individual graphics. For example:

```
\includegraphics[draft,width=1.2in]{box.eps}
```

8 Specifying Height and/or Width in graphicx

The graphic's height and/or width can be specified, resulting in the following combinations:

- If neither the height nor the width is specified, the EPS graphic is included with its natural size (the size specified by the BoundingBox).
- If the height is specified and the width is not specified, the EPS graphic is included with the specified height and a width such that its height/width aspect ratio remains the same.
- If the width is specified and the height is not specified, the EPS graphic is included with the specified width and a height such that its height/width aspect ratio remains the same.
- If both height and width are specified:
  - If the `keepaspectratio` option is *not* specified, the EPS graphic is scaled anamorphically to fit both the specified height and width.
  - If the `keepaspectratio` option is specified, the graphic is made as large as possible such that its aspect ratio remains the same and the graphic does not exceed either the specified height or width.

The following LaTeX command makes the included graphic as wide as the text:

```
\includegraphics[width=\textwidth]{box.eps}
```

The following LaTeX command makes the included graphic 80% as wide as the text:

```
\includegraphics[width=0.80\textwidth]{box.eps}
```

The following commands make the width of the included graphic 2 inches less than the width of text:

```
\newlength{\epswidth}
\setlength{\epswidth}{\textwidth}
\addtolength{\epswidth}{-2.0in}
\includegraphics[width=\epswidth]{box.eps}
```

If the calc package is available, this is shortened to:

```
\newlength{\epswidth}
\setlength{\epswidth}{\textwidth -2.0in}
\includegraphics[width=\epswidth]{box.eps}
```

The `\newlength` command only needs to be issued once. Subsequent graphics can be scaled without reissuing the `\newlength` command. The length name `\epswidth` is not special. Any other name (which isn't already used by LaTeX) could have been used. The calc package with the 12/95 graphicx package shortens this further to:

```
\includegraphics[width=\textwidth-2.0in]%
                                 {box.eps}
```

8.1 Problems with Specifying Height

Users must be careful when using the `height` option. When users want to specify an object's "height", they often mean the overall height which is set by the `totalheight` option and not the `height` option. If the `height` option is mistakenly used instead of `totalheight`, the results may or may not be bad:

- If the object happens to have a zero depth (see the left box in Figure 2) the `totalheight` is the same as the `height` and everything works fine.
- If the object has non-zero depth (see the middle box in Figure 2) the object is scaled such that the object's `height` is as large as the desired `totalheight`, making the object too large.
- If the object has zero height (see the right box in Figure 2) an "Arithmetic overflow" (divide-by-zero) error occurs. This happens because LaTeX calculates the scaling factor as

$$\text{scaling} = \frac{\text{requested height}}{\text{height}}$$

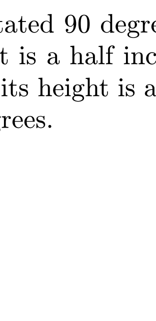
Note that the LaTeX2.09 `\psfig` command and early versions of `\includegraphics` only have a `height` option.

9 Rotation with Scaling in graphicx

Since the `\includegraphics` options are interpreted from left to right, the order in which the angle and size are specified makes a difference. For example:

```
\begin{center}
  \includegraphics[angle=90,%
          totalheight=0.5in]{box.eps}
  \includegraphics[totalheight=0.5in,%
                   angle=90]{box.eps}
\end{center}
```

produces:



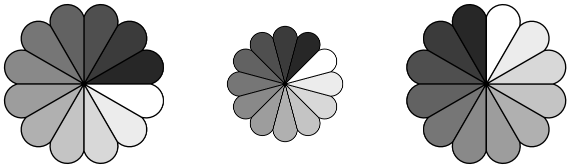
The first box is rotated 90 degrees and then scaled such that its height is a half inch. The second box is scaled such that its height is a half inch and then it is rotated 90 degrees.

9.1 Scaling of Rotated Graphics

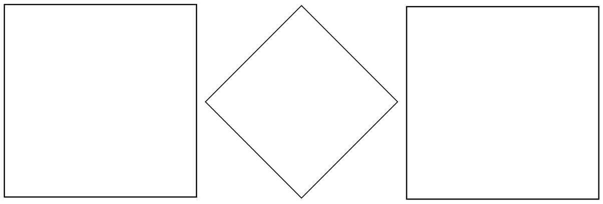
When the height or width of a graphic is specified, the specified size is not the size of the graphic but rather of its BoundingBox. This distinction is especially important in order to understand the scaling of rotated graphics. For example:

```
\begin{center}
  \includegraphics[totalheight=1in]%
                           {rosette.eps}
  \includegraphics[angle=45,totalheight=1in]%
                           {rosette.eps}
  \includegraphics[angle=90,totalheight=1in]%
                           {rosette.eps}
\end{center}
```

produces:



Although it may seem strange that the graphics have different sizes, it makes sense after viewing the BoundingBoxes:



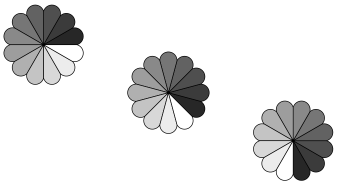
Each graphic is scaled such that its rotated BoundingBox is 1 inch tall.

9.2 Alignment of Rotated Graphics

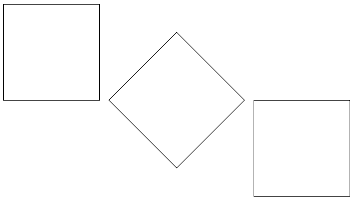
When graphics are rotated, the objects may not align properly. For example:

```
\begin{center}
  \includegraphics[totalheight=0.5in]%
                            {rosette.eps}
  \includegraphics[totalheight=0.5in,%
                  angle=-45]{rosette.eps}
  \includegraphics[totalheight=0.5in,%
                  angle=-90]{rosette.eps}
\end{center}
```

produces:



Again, this is better illustrated by the BoundingBoxes:



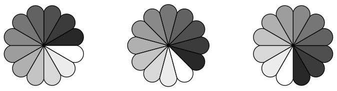
In this case, the objects' reference points (original lower-left corners) are aligned on a horizontal line. If it is desired to instead have the centers aligned, the `minipage` environment can be used:

```
\begin{center}
  \begin{minipage}[c]{0.625in}
    \centering
    \includegraphics[totalheight=0.5in,%
                     angle=0]{rosette.eps}
  \end{minipage}
  \begin{minipage}[c]{0.625in}
    \centering
    \includegraphics[totalheight=0.5in,%
                   angle=-45]{rosette.eps}
  \end{minipage}
  \begin{minipage}[c]{0.625in}
    \centering
    \includegraphics[totalheight=0.5in,%
                   angle=-90]{rosette.eps}
  \end{minipage}
\end{center}
```

However, an easier solution uses the `\rotatebox` command to rotate the graphic about its center:

```
\begin{center}
  \includegraphics[totalheight=0.5in]%
                             {rosette.eps}
  \rotatebox[origin=c]{-45}{%
     \includegraphics[totalheight=0.5in]%
                             {rosette.eps}}
  \rotatebox[origin=c]{-90}{%
     \includegraphics[totalheight=0.5in]%
                             {rosette.eps}}
\end{center}
```

This aligns the centers of the graphics:



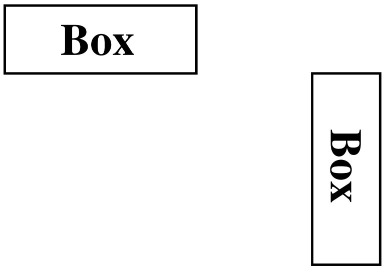
If the 12/95 version of graphicx is used, the `origin` option can be used in `\includegraphics`:

```
\begin{center}
  \includegraphics[totalheight=0.5in]%
                                {rosette.eps}
  \includegraphics[totalheight=0.5in,%
             origin=c,angle=-45]{rosette.eps}
  \includegraphics[totalheight=0.5in,%
             origin=c,angle=-90]{rosette.eps}
\end{center}
```

Similarly, the commands:

```
\begin{center}
    \includegraphics[width=1in]{box.eps}
    \hspace{0.5in}
    \includegraphics[width=1in,angle=-90]%
                                 {box.eps}
\end{center}
```

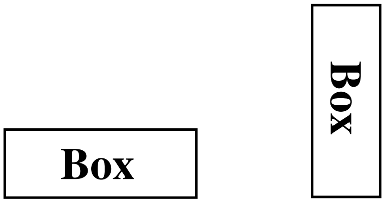
produce:



while the following commands:

```
\begin{center}
  \includegraphics[width=1in]{box.eps}
  \hspace{0.5in}
  \rotatebox[origin=br]{-90}{%
    \includegraphics[width=1in]{box.eps}}
\end{center}
```

align the bottoms of the graphics:



If the 12/95 version of graphicx is used, the `origin` option can be used in `\includegraphics`:

```
\begin{center}
  \includegraphics[width=1in]{box.eps}
  \hspace{0.5in}
  \includegraphics[width=1in,origin=br,%
                   angle=-90]{box.eps}}
\end{center}
```

10 Compressed and Non-EPS Graphics Files

The commands `\DeclareGraphicsRule` and `\DeclareGraphicsExtensions` control how LaTeX deals with the files specified in `\includegraphics` commands.

When using `dvips`, users can specify an operation to be performed on the file before it is inserted. By making this operation a decompression command, compressed graphics files can be used. By making this a graphics-conversion command, non-EPS graphics files can be used. **Since `dvips` is currently the only DVI-to-PS converter with this capability, everything in this section requires `dvips`.**

`\DeclareGraphicsRule` has a 'command' argument; this is only usable on an operating system that supports pipes. Without piping, the decompression or conversion cannot be done on-the-fly and the user must store all graphics as uncompressed EPS files.

Depending on the system defaults, users may need to pass the `dvips` option to the graphicx package. This can be done by either specifying the `dvips` global option in the `\documentclass` command:

```
\documentclass[dvips,11pt]{article}
```

or by specifying the `dvips` option when loading the package using the `\usepackage` command:

```
\usepackage[dvips]{graphicx}
```

Since specifying the `dvips` as a global option passes it to all packages, it is generally preferred.

10.1 The DeclareGraphicsRule Command

The `\DeclareGraphicsRule` command specifies how `\includegraphics` should treat files depending on their extensions. Multiple `\DeclareGraphicsRule` commands may be issued. The syntax is:

```
\DeclareGraphicsRule{ext}{type}%
                    {sizefile}{command}
```

See Table 4 for details of the arguments to the command. For example, the following command:

```
\DeclareGraphicsRule{.eps.gz}{eps}{.eps.bb}%
                             {`gunzip -c #1}
```

specifies that any file with a `.eps.gz` extension is treated as compressed `eps` file. It also specifies that the BoundingBox information is stored in the file with a `.eps.bb` extension, and that `gunzip -c` command uncompresses the file. (Since LaTeX cannot read BoundingBox information from a compressed file, the BoundingBox line must be stored in an uncompressed file.)

Since the following graphics rules are defined by default in `dvips.def`, users generally do not need to use the `\DeclareGraphicsRule` command:

```
\DeclareGraphicsRule{.eps}{eps}{.eps}{}
\DeclareGraphicsRule{.ps}{eps}{.ps}{}
\DeclareGraphicsRule{.pz}{eps}{.bb}%
                          {`gunzip -c #1}
\DeclareGraphicsRule{.eps.Z}{eps}{.eps.bb}%
                          {`gunzip -c #1}
\DeclareGraphicsRule{.ps.Z}{eps}{.ps.bb}%
                          {`gunzip -c #1}
\DeclareGraphicsRule{.eps.gz}{eps}{.eps.bb}%
                          {`gunzip -c #1}
\DeclareGraphicsRule{.ps.gz}{eps}{.ps.bb}%
                          {`gunzip -c #1}
\DeclareGraphicsRule{.pcx}{bmp}{}{}
\DeclareGraphicsRule{.bmp}{bmp}{}{}
\DeclareGraphicsRule{.msp}{bmp}{}{}
\DeclareGraphicsRule{*}{eps}{*}{}
```

Table 4: \DeclareGraphicsRule Arguments

ext	The file extension.
type	The graphics type for that extension.
sizefile	The extension of the file which contains the BoundingBox information for the graphics. If this option is blank {}, the size information must be specified by an \includegraphics option.
command	The command to be applied to the file (often left blank {}). The command must be preceded by a single backward quote (not to be confused with the more common forward single quote).

The first two commands define the .eps and .ps extensions as eps files. The next five commands define extensions for compressed eps files. The next three commands define extensions for bitmaps (see section 10.3.1). The last command defines any other suffix as an eps file.

For example, to compress the EPS file file.eps, the BoundingBox line must first be extracted and stored in file.eps.bb. The EPS file can then be compressed by the gzip file.eps command (maximise compression by using gzip -9 file.eps). The old epsfig package came with epsbb, a perl script which creates the BoundingBox file and then compresses the EPS file. epsbb is still available from CTAN.

10.2 The DeclareGraphicsExtensions Command

The \DeclareGraphicsExtensions command tells LaTeX which extensions to try if a user specifies a file with no extension in the \includegraphics command. The following graphic extensions are defined by default in dvips.def:

```
\DeclareGraphicsExtensions{.eps,.ps,%
                  .eps.gz,.ps.gz,.eps.Z}
```

With the above graphics extensions specified, the command \includegraphics{file} makes LaTeX first look for file.eps, then for file.ps, then for file.eps.gz, etc., until a file is found.

The \DeclareGraphicsExtensions command allows the graphics to be specified with

```
\includegraphics{file}
```

instead of

```
\includegraphics{file.eps}
```

The first syntax has the advantage that if you later decide to compress file.eps, you do not need to edit the LaTeX file.

10.3 Including Non-EPS Graphic Files

While it is easy to insert EPS graphics into LaTeX documents, it is not as straightforward to insert non-EPS graphics (GIF, TIFF, JPEG, PICT, etc.). A simple solution is to find out if the application which generated the non-EPS graphic also generates EPS output. If not, a graphics conversion program (such as ImageMagick, xv, netpbm, pbmplus) must be used to convert the graphics to PostScript.

Since a non-EPS graphics file may be smaller than the corresponding EPS file, it may be desirable to keep the graphics in a non-EPS format and convert them to PostScript when the DVI file is converted to PostScript. If dvips is used, this on-the-fly conversion can be specified by the command option in \DeclareGraphicsRule. For example, to use on-the-fly conversion to insert file.gif into a LaTeX document, one needs to:

1. Find a GIF-to-PS conversion program (assume it's called gif2ps)
2. One needs to create a .bb file which specifies the natural size of the file.gif graphics. To do this, convert file.gif to PostScript and:
   (a) If the Postscript file is EPS, save the BoundingBox line in file.bb
   (b) If the Postscript file is not EPS, determine the appropriate BoundingBox (see section 3) and store it in file.bb
3. Keep file.gif and delete the PostScript file.
4. Enter \DeclareGraphicsRule{.gif}{eps}%
   {.bb}{‘gif2eps #1} in the LaTeX document.

When the command \includegraphics{file.gif} is issued, LaTeX will read the BoundingBox from file.bb and will also tell dvips to use gif2eps to convert file.gif into PostScript.

10.3.1 Direct Support for Non-EPS Graphics

It is often requested that LaTeX and dvips support the direct inclusion of non-EPS graphic formats, making it as easy as inserting EPS files. While this would be convenient, there unfortunately are some problems which complicate things.

For example, most non-EPS graphic formats use binary files which cannot be read by TeX, which prevents LaTeX from determining the size of the non-EPS graphics. Furthermore, supporting non-EPS graphics would also require `dvips` to incorporate graphics conversion capabilities (GIF-to-PS, TIFF-to-PS, etc.). This would not only require a lot of programming, it would also require more maintenance in the future.

Rather than directly incorporating graphics conversion routines, `dvips` provides a mechanism of calling external conversion programs. This mechanism can be accessed from LaTeX by use the the command argument of `\DeclareGraphicsRule`. This has the benefit of being more flexible than direct support, and since it keeps the graphics conversion uncoupled from the DVI-to-PS conversion, users are free to choose their own graphics conversion program.

While LaTeX and `dvips` generally do not support the direct inclusion of non-EPS graphics, there are some exceptions:

1. If `dvips` is compiled with `-Demtex`, it supports some EmTeX `\special` commands, allowing it to include PCX, BMP, or MSP bitmaps.
2. Some commercial versions of LaTeX support non-EPS graphics:
   (a) *Textures* for the Macintosh supports PICT graphics.
   (b) Y&Y's TeX package for Windows includes the DVI-to-PS converter `DVIPSONE` which supports TIFF files. However, TeX cannot read the binary TIFF files, preventing LaTeX from reading the TIFF tags the same way it reads EPS BoundingBox information. Since LaTeX cannot determine the natural size of TIFF graphics, the user must still use a `.bb` file or specify the `bb` parameters explicitly in the optional argument of the `\includegraphics` command.

Check your documentation or contact the company's customer service for the correct syntax.

References

[1] David P. Carlisle. Packages in the 'graphics' bundle. Available from CTAN as `grfguide.tex` or `grfguide.ps`.

[2] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994.

[3] Helmut Kopka and Patrick Daly. *A Guide to LaTeX 2ε.* Addison-Wesley, Reading, Massachusetts, 1995.

[4] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, second edition, 1994.

◇ Keith Reckdahl
Stanford University
Box 9030
Palo Alto, CA 94309
USA
`reckdahl@leland.stanford.edu`

Macros

That ol' devil `\expandafter`

Don Hosek

Every TEX hacker knows that `\expandafter` is an essential part of one's macro vocabulary. Since TEX is a macro-based language, controlling the order of macro expansion is essential to making certain effects work.

However, because in practice `\expandafter` commands tend to come in swarms, it is often difficult to follow all but the most common idioms when looking at TEX macro code.

I recently was faced with this problem in the development of the qstyle core macros which I use in developing style and class files. The truth of the matter is that there is *never* code written which is not modified later. I consider it an essential matter to document every bit of code I write, when I write it, or I'll never be able to maintain it.

The qstyle code contains some relatively simple idioms like

```
\expandafter\expandafter\expandafter
   \A
\expandafter
   \B
   \C
```

which would cause the order of expansion of `\A`, `\B` and `\C` to be reversed.[1]

[1] Highly recommended reading on this topic would include Stephen Bechtolsheim's article in *TUGboat* Vol. 9, No. 1, which appears in a modified form in his book, *TEX in Practice*.

But for non-standard expansions, say, to expand `\B`, then `\C`, then `\A`,[2] the code may become a little more convoluted to say the least. In this case, an unambiguous representation of the order of expansion comes in handy. Towards this end, I came up with an indentation-based approach to getting the job done. The basic principle is to have the first level of indentation indicate one set of macros expanded, and keep indenting to complete the job.The first macro I used this on was the `\QNameLetName` macro which allows me to take the names of two control sequences and `\let` the equivalent `\csname` of the first to the `\csname` of the second. The macro is much simpler if the first `\csname` is expanded before the second since we then have only one token to skip over.[3] In traditional indentation schemes, the code would be hard to follow, but using an indentation based on expansion order, it becomes much easier to follow.

```
\def\QNameLetName#1#2{%
   \expandafter
      \expandafter
   \expandafter
         \let
   \expandafter
      \expandafter
   \csname#1\endcsname
      \csname#2\endcsname
}
```

Annex

During the review process, it was pointed out by Victor Eijkhout that there is a simpler solution to the programming problem above:

```
...
\expandafter\let
  \csname #1\expandafter\endcsname
     \csname #2\endcsname
...
```

Note that while this is a simpler and more efficient solution, the argument above about intelligent indentation stands.

◇ Don Hosek
Quixote Digital Typography
555 Guilford
Claremont, CA 91711
dhosek@quixote.com

[2] By the way, this can only be done if `\B` expands to a single token.

[3] One of the joys of TeX: Sometimes when we expand something, we end up with *fewer* tokens than when we started.

PPCHTEX: typesetting chemical formulas in TEX

J. Hagen and A.F. Otten

Abstract

This article is about a package for typesetting chemical formulas. The primary interface of this package is in the Dutch language. Because PPCHTEX has a multilingual interface, all commands and keywords can be toggled to English. The Dutch version of this article is published in *NTG*'s MAPS (94.2) and is translated to English by H. de Weert.

1 Introduction

The macro-package PPCHTEX can be used to typeset chemical formulas. The macros are based on PICTEX, a macropackage that was created to facilitate the drawing of graphics and other line diagrams. PICTEX was brought to the public domain by M.J. Wichura. We are considering a second implementation, using T. van Zandt's PSTRICKS.

The macros can be used within different TEX-environments because they only depend on Knuth's plain TEX. In addition, some general macros of our ConTEXt-library are utilized. Macros are written in such a way that further development is quite easy.

At first, macros are created to typeset chemical structure formulas. Moreover, reaction mechanisms can be reflected. Chemical structures can be typeset in different sizes and comparable formulas can be linked optically. Structures of frequent occurrence can be predefined and recalled.

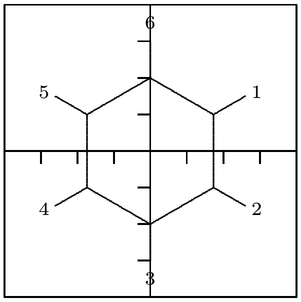
During the development of the macros, processing speed is subordinated to flexibility, simplicity and quality. No use has been made of the mechanism (available in PICTEX) to store parts of figures in a file. It turned out that this mechanism does not produce a gain of time.

The macros are still being developed. For example, the mechanism to place texts still has to be refined and some structures like `CHAIR` *still have to be added.*

2 Structures

The number of commands that is used to typeset chemical structure formulas is reduced to four.[1] In the following example all of these commands are used.

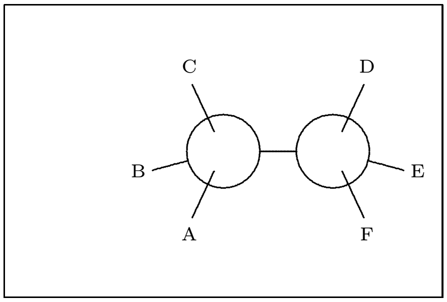
[1] The concept structure in this manual only refers to the chemical structure. It is not related to the structure of the text that is used to typeset the formula.



Example 1

```
\setupchemical[axis=on,border=on]
\startchemical
  \chemical[SIX,B,R,RZ][1,2,3,4,5,6]
\stopchemical
```

Different features of the typesetting can be set up with `\setupchemical`. If something is set up in this way, the setups are valid for all the following formulas.[2]

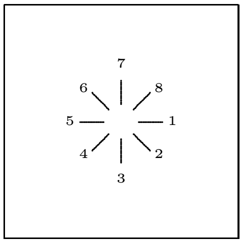


Example 2

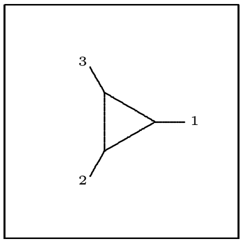
```
\startchemical[border=on,width=6000]
  \chemical[CARBON,CB1][A,B,C,D,E,F]
\stopchemical
```

As can be seen from both examples, `\chemical` is the central command. This command, that can be typed many times within a `\start`-`\stop`-pair, gets one or two arguments. These arguments are given between `[ ]`. The first argument refers to the bonds that are to be drawn. The second argument contains the atoms or molecules that are to be reflected. Text is typesetted in a mathematical mode, so everything that is normally allowed between `$ $` can be given.

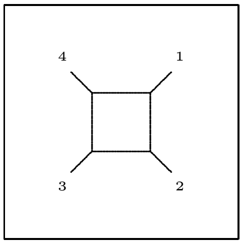
We work out the first example. First of all the keyword `SIX` is given. By using this word we can indicate that we want to draw a sixring structure. In the same way we can use the keywords `ONE`, `THREE`, `FOUR`, `FIVE`, `NEWMAN` and `CARBON`.



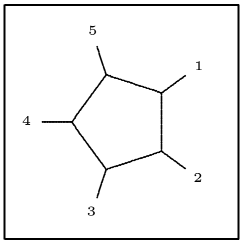
One



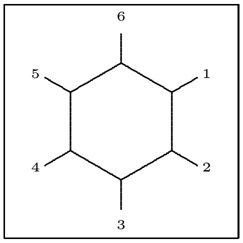
Three



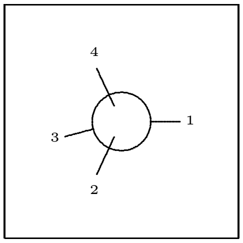
Four



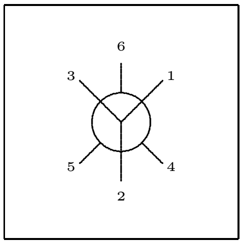
Five



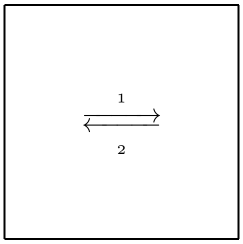
Six



Carbon



Newman



Symbol

Within these structures chemical bonds between C-atoms can be indicated in a comparable way. For instance, in this example we use `B` and `R`. Bonds are numbered and can be indicated in different ways:

```
\chemical[SIX,B1,B2,B3,B4,B5,B6]
\chemical[SIX,B135]
\chemical[SIX,B1..5]
```

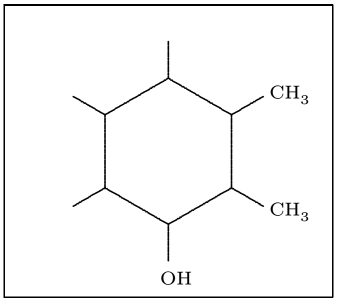
These commands create parts of a sixring structure. `R` enables us to add substituents to the sixring structure. The command `R` draws the beginning of a bond with a substituent from an angular point in the sixring structure (∠ 120°). The concerning angular point is indicated with a number.

```
\chemical[SIX,B1..6,R1..6]
```

The above mentioned command only places bonds to substituents. Substituents themselves are indicated with `RZ`. Therefore in this case, numbers

[2] Obviously the scope can be confined by using { } and the grouping macros `..group`. The setups can also be given immediately after `\startchemical`. In that case the setups are applied to one formula.

are being used to mark the position. In the second optional argument substituents are given as text.



Example 3

```
\startchemical[border=on,width=4500]
\chemical[SIX,B1..6,R1..6,RZ1..3][CH_3,CH_3,OH]
\stopchemical
```

If the second argument is omitted, no text is placed, so the command `RZ1..3` has no effect.

3 Definitions

It is possible to build a library of structures. As we wish, we can recall these structures at a later point of time and provide them with extra components. Furthermore they can serve as building blocks for more complex structures. Structures can be predefined with the TEX-primitive `\def`.

If a structure, for example `[SIX,B,R,RZ]`, is often used, it is practical to predefine this structure.

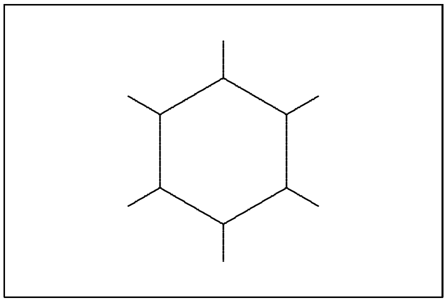
```
\def\sixring{\chemical[SIX,B,R,RZ]}
```

Instead of `\def` the following command can be used. In this case an already existing definition will be announced.

```
\definechemical[sixring]
  {\chemical[SIX,B,R,RZ]}
```

Although both ways of defining are allowed, the second way is more robust. Protective measures are taken to avoid conflicts with existing commands.

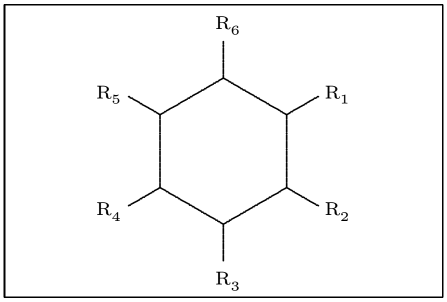
The command `\chemical[sixring]` provides a sixring structure without substituents. No second argument is given.



Example 4

```
\definechemical[sixring]
  {\chemical[SIX,B,R,RZ]}

\startchemical[border=on,width=6000]
  \chemical[sixring]
\stopchemical
```

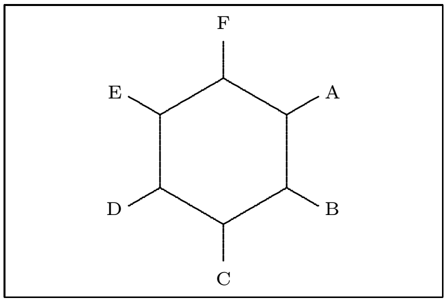
If we want to add six substituents, we have to carry out the following actions:



Example 5

```
\definechemical[sixring]
  {\chemical[SIX,B,R,RZ]}

\startchemical[border=on,width=6000]
  \chemical[sixring]%
           [R_1,R_2,R_3,R_4,R_5,R_6]
\stopchemical
```

The sixring structure can also be defined without substituents (`RZ`). In this case no substituents are expected if the command `\chemical` is given. Even now substituents can be placed, as is shown by the following example.



Example 6

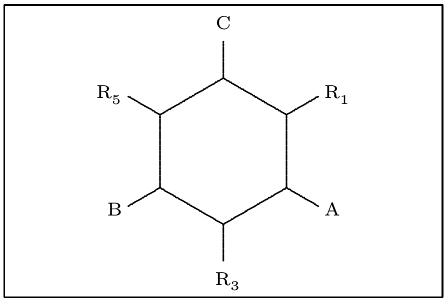
```
\definechemical[sixring]
  {\chemical[SIX,B,R]}

\startchemical[border=on,width=6000]
  \chemical[sixring,RZ][A,B,C,D,E,F]
\stopchemical
```

Essentially the number of possibilities is unlimited. One should be aware of the fact that the atoms and molecules of the second argument are raised in the sequence of the first argument.

In a definition atoms and molecules (texts) can also be placed.

```
\definechemical[sixring]
  {\chemical[SIX,B,R,RZ135][R_1,R_3,R_5]}
```

So in this definition always three substituents are added. If we decide to add more substituents, we have to explicitly state that we are dealing with a sixring structure (SIX).

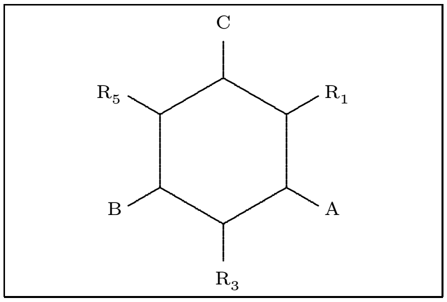


Example 7

```
\definechemical[sixring]
{\chemical[SIX,B,R,RZ135][R_1,R_3,R_5]}

\startchemical[border=on,width=6000]
  \chemical[sixring,SIX,RZ246][A,B,C]
\stopchemical
```

So in definitions, the command `\chemical` has a global character and the command `\chemical[][]` has a local character. The idea behind this is that in the first case a series of commands is inserted and in the second case a complete independent structure is inserted.

In a definition the command `\chemical` can occur more than once. The last example can also be recalled with:



Example 8

```
\definechemical[sixring]
  {\chemical[SIX,B,R,RZ135][R_1,R_3,R_5]
   \chemical[SIX,RZ246]}

\startchemical[border=on,width=6000]
  \chemical[sixring][A,B,C]
\stopchemical
```

If PPCHTEX makes mention of an unknown command, one has probably forgotten to type a structure command, like SIX or FIVE.

4 Bonds

In this section we show the bonds that can be found in the different chemical structures. The meaning of the commands will be explained by the reviews that are stated further in this article.

In the left column the complete bonds are shown, in the right column only the shortened bonds. Due to these shortened bonds, atoms and molecules can be attached to a bond. Bonds can be shortened on both sides, left (-) as well as right (+).

B	Bond	SB	Single Bond
		-SB	Left Single Bond
		+SB	Right Single Bond

Table 1: Saturated bonds

A bond can be followed by one or more numbers or a range, for instance: B1, B135 and B1..5. If all bonds are necessary, only B can be given.

In a ring structure an extra bond can be given and furthermore double or triple bonds can be introduced between atoms and molecules.

EB	Extra Bond	DB	Double Bond
		TB	Triple Bond

Table 2: Unsaturated bonds

A bond, in a sixring structure for example, can be shortcut. In this case the atom is given that has to be omitted, therefore a circle can be drawn in a sixring structure.

S	Shortcut	C	Circle

Table 3: Special bonds

Substituents can be attached to the angular points and depending on the presence of atoms and molecules the bonds can be short or long.

It is possible to bind substituents to the structure by double bonds.

Text can be linked to bonds. These texts are collected from the second set behind `\chemical` in the sequence that is given.

The atoms—/—molecules are numbered clockwise. In this case, combinations are allowed. With Z0 (z zero) a text can be placed in the middle of a structure.

R	Radical
SR	Single Radical
-R	Left Radical
-SR	Single Left Radical
+R	Right Radical
+SR	Single Right Radical

Table 4: Bonds to substituents

ER	Extra Radical	DR	Double Radical

Table 5: Double bonds to substituents

While positioning the atoms and molecules in the text, their (possible) dimensions are taken into account. In this case the width of C and the height of C^n_m play a prominent role. However, this mechanism can still be refined.

5 Combinations

Structures can be combined to complex compounds and can be moved to other structures by using `MOV`, `ROT`, `ADJ` and `SUB`.

These four commands have another effect within the different structures. For example, the angle used to rotate at `\chemical[FIVE,ROT1,B]` differs from the angle that is used at `\chemical[SIX,ROT1,B]`.

In addition, within `CARBON` it is possible to mirror a structure. This can be done with `MIR`.

The direction of a displacement or the amount of the rotation is indicated by a number. Since these commands are closely related to the actual structure, they must be given before bonds and texts are drawn. It makes a difference whether one gives the command `\chemical[FIVE,B,ROT1,R]` or `\chemical[FIVE,ROT1,B,R]`. The first call delivers an unwanted result.

Z	Atom	RZ	Radical Atom
		-RZ	Left Radical Atom
		+RZ	Right Radical Atom

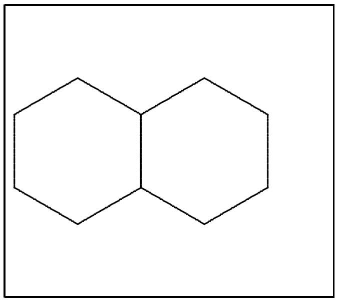
Table 6: Atoms and molecules (radicals)

MOV	Move	move the same kind of structure in the direction of a bond
ADJ	Adjacent	move a different kind of structure in the direction of the x- or y-axis, linked to a bond
SUB	Substitute	move a structure relative to another one in the direction of the x- or y-axis
ROT	Rotate	rotate a structure

Table 7: Displacements and rotations.

MIR	Mirror	mirror a structure

Table 8: Mirroring

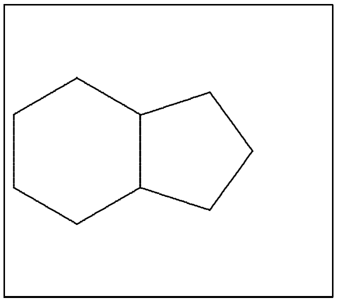


Example 9

```
\startchemical[border=on,width=4500,right=3500]
  \chemical[SIX,B,MOV1,B]
\stopchemical
```

Successively a sixring structure is drawn: `SIX,B`, a displacement is realized in the direction of bond 1: `MOV1`, and a second sixring structure is drawn: `B`. A displacement with `MOV` concerning a sixring structure can be realized in six directions, as opposed to a displacement with `ADJ`, which is realized in the four axis-directions (x, $-x$, y, $-y$). In a sixring structure some of these displacements coincide. The above example also could have been achieved with: `[SIX,B,ADJ1,B]`.

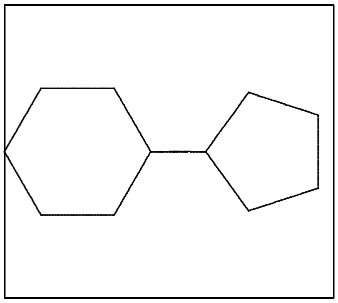
It is also possible to combine different structures. For instance, `SIX` can be linked to a structure `FIVE`. The mechanism that is responsible for this linking is for the greater part hidden from the user. In the following example a sixring structure is successively drawn: `SIX,B`, a displacement along the positive x-axis is achieved: `ADJ1`, and a rotated fivering structure is drawn: `FIVE,ROT3,B`.



Example 10

```
\startchemical[border=on,width=4500,right=3500]
  \chemical[SIX,B,ADJ1,FIVE,ROT3,B]
\stopchemical
```

A transition to a connected structure can be achieved with ADJ. To get a good connection, one of the two structures have to be rotated with ROT. If a structure is not directly linked, but through a bond, one uses SUB. Rotations are made in steps of 90 degrees, clockwise. Displacements with ADJ and SUB are achieved in the four axis-directions.



Example 11

```
\startchemical[border=on,width=4500,right=3500]
  \chemical[SIX,ROT2,B,R6,SUB1,FIVE,B,R4]
\stopchemical
```

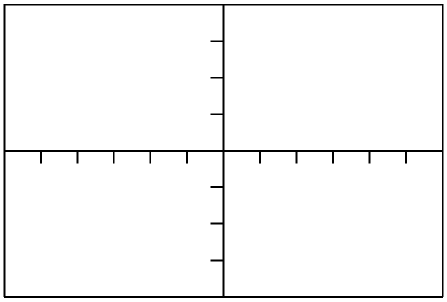
We can therefore conclude that the sequence of the given commands is very important. An obvious sequence of commands is as follows:

```
\chemical
 [structure,                  % SIX,FIVE,...
  bonds inside the structure, % B,C,EB,...
  bonds outside the structure,% R,DR,...
  locations of atoms,         % Z
  locations of substituents]  % RZ,-RZ,...
 [atoms,
  substituents]
```

As a rule, the connection of structures is reduced to some translations and rotations. Although it may not seem so, a certain systematic is enclosed. In fact, the process could be simplified. The automation that was already achieved in former versions has been undone: it turned out that 'hidden' rotations induce misunderstandings with regard to the place of bonds. Furthermore, it is easier to provide a structure that is not rotated with bonds, atoms and molecules than to provide a rotated structure. It is better to define the parts of a complex structure first, possibly with translations, and to rotate the complete complex structure later.

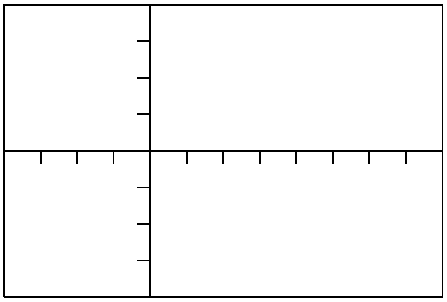
6 Axis

Structures are typeset in a bounded space, for convenience indicated by axis. The dimensions of the axis and the location of the origin can be defined in the setup. In addition the axis can be made visible (for the sake of the location in the text) and a border can be drawn.



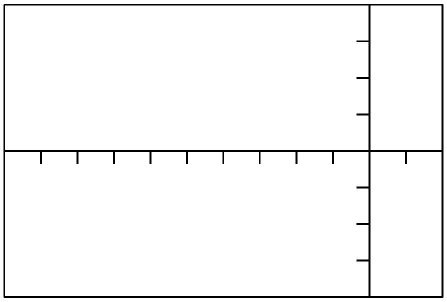
Example 12

```
\startchemical
  [axis=on,
   width=6000,height=4000]
  ...
  ...
\stopchemical
```



Example 13

```
\startchemical
  [axis=on,
   left=2000,right=4000]
  ...
  ...
\stopchemical
```

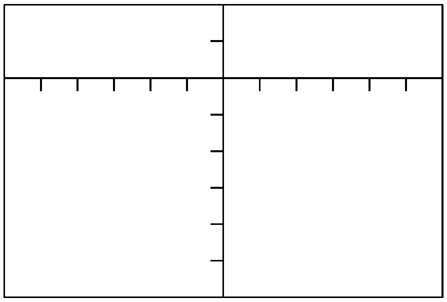


Example 14

```
\startchemical
  [axis=on,
   width=6000,right=1000]
  ...
  ...
\stopchemical
```

The dimensions of the structure are determined by the dimensions of the axis. However, if at least one of following two options `width=fit` and `height=fit` is given, the dimensions of the total structure are determined by the real dimensions. Whatever is chosen is depending on the way structures are placed in the text. Side by side, on top of each other, etc. Example 12 shows the standard setups.

Within a `\start`-`\stop`-pair PICTEX-macros can be used. Of course, some caution must be taken into account.



Example 15

```
\startchemical
  [axis=on,
   width=6000,top=1000,bottom=3000]
  ...
  ...
\stopchemical
```

7 Setups

The behavior of the macros and the layout of the formulas can be adapted to one's personal needs. After both `\startchemical` and `\setupchemical` setups can be given.

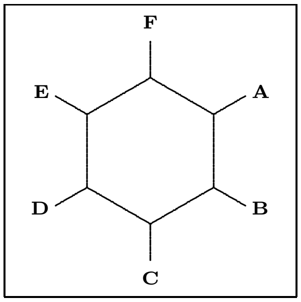
variable	**values**	**default**
width	number	4000
height	number	4000
left	number	
right	number	
top	number	
bottom	number	
resolution	number	see manual
corps	8pt 9pt 10pt etc.	see manual
style	`\rm` `\bf` etc.	see manual
scale	number small medium big	medium
size	small medium big	medium
status	start stop	start
option	test	
axis	on off	off
border	on off	off
alternative	1 2	1
offset	HIGH LOW	LOW

Table 9: Setups for structures

The axis reaches from −2000 to +2000, in height as well as in width. The point `Z0` is situated at (0,0). Other setups can be set up with `left`, `right`, `above` and/or `under` in combination with `width` and `height`.

The dimensions of the characters can be set up with `size`. In doing so, the TEX-primitives `\textsize`, `\scriptsize` and `\scriptscriptsize` are used. The dimensions of the structure itself can be set up with `scale` (`1..1000`). The scale is also determined by `corps`. The keywords `small`, `medium` and `big` are attuned.

In the mathematical mode commands like `\rm`, `\bf` and `\sl` can be utilized in TEX as well as in ConTEXt. PPCHTEX by default uses `\rm`. An alternative command can be set up with `style`. For instance in example 16 the substituents are typeset **bold**.



Example 16

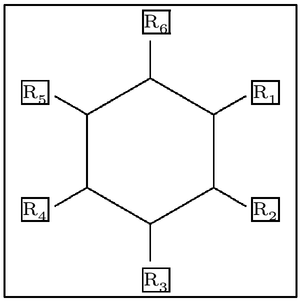
```
\startchemical[border=on,style=\bf]
```

```
  \chemical[SIX,B,R,RZ][A,B,C,D,E,F]
\stopchemical
```

For the time being `style` is valid for chemical formulas in the text as well as in a figure. The sub- and superscripts also change, as can be seen from: CH_4, CH_4 and CH_4. Successively the setups are: `\rm`, `\bf` and `\sl`. When formulas are typeset in italic (`\it`), baseline-distances can be greater than normal. Within ConTEXt bold-slanted (`\bs`) and bold-italic (`\bi`) are also available. All these commands automatically adjust to the actual style (`\ss`, `\rm`, `\tt`): CH_4, CH_4 etc.

The time-consuming calculations can be short-circuited with `status`. The variables `border` and `axis` speak for themselves. A border round a text can be drawn with `option=test`. By doing so, one can see how things are aligned. The quality of the lines is set up with `alternative`. By default, PICTEX uses a 5 point period to draw the lines. As one chooses alternative 2 smaller points are used. Therefore thinner lines are drawn.

The offset refers to the position of the sub- and superscripts. By using `HIGH` the subscripts are positioned high (H_2O). Obviously, with `LOW`, the subscripts are positioned a little lower (H_2O).

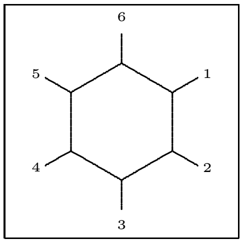


Example 17

```
\startchemical[border=on,option=test,%
               alternative=2]
 \chemical[SIX,B,R,RZ][R_1,R_2,R_3,R_4,R_5,R_6]
\stopchemical
```

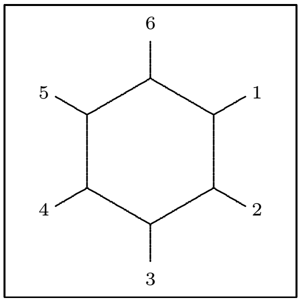
8 Dimensions

A structure can be shown in different sizes. Dimensions can be set up with `size` and `scale`.



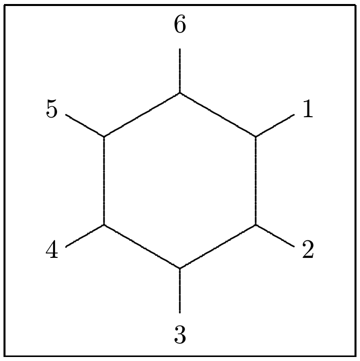
Example 18

```
\startchemical[scale=small,size=small]
  \chemical[SIX,B,R,RZ][1,2,3,4,5,6]
\stopchemical
```



Example 19

```
\startchemical[scale=medium,size=medium]
  \chemical[SIX,B,R,RZ][1,2,3,4,5,6]
\stopchemical
```



Example 20

```
\startchemical[scale=big,size=big]
  \chemical[SIX,B,R,RZ][1,2,3,4,5,6]
\stopchemical
```

Scaling can take place between 1 and 1000. The values belonging to the keywords `small`, `medium` and `big` are attuned.

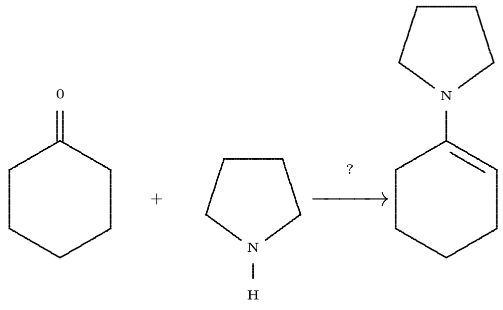
9 Symbols

To typeset reaction equations some symbols are available. In the following figure an equation is shown. This equation is defined as follows:

```
\setupchemical
  [width=fit,
   height=5500,
   under=1500]
\hbox
  {\startchemical
     \chemical[SIX,B,ER6,RZ6][0]
   \stopchemical
   \startchemical
```

```
      \chemical[SPACE,PLUS,SPACE]
   \stopchemical
   \startchemical
      \chemical[FIVE,ROT4,B125,+SB3,-SB4,Z4,%
               SR4,RZ4][N,H]
   \stopchemical
   \startchemical
      \chemical[SPACE,GIVES,SPACE][?]
   \stopchemical
   \startchemical
      \chemical[SIX,B,EB6,R6,SUB4,FIVE,ROT4,%
                B125,+SB3,-SB4,Z4][N]
   \stopchemical}
```

The command \hbox is necessary to position the structures behind each other. The symbols GIVES and PLUS speak for themselves. Extra space can be achieved with SPACE.



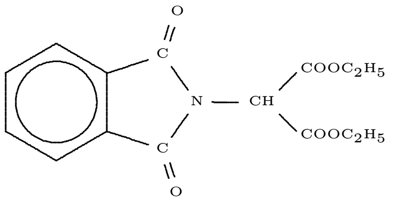
An equilibrium can be shown with EQUILIBRIUM. A text can be set above GIVES and EQUILIBRIUM. In the example this text is a question mark.

10 Special features

By using ONE, ZO can consist of more than one atom. In this case, the reserved space is insufficient. If more space is needed for the command ZO, the bonds 1, 2 and 8 can be moved with the command OFF, which means 'offset'. Below, an example of the use of this command is given.

```
\startchemical[width=fit]
  \chemical
    [SIX,B,C,ADJ1,
     FIVE,ROT3,SB34,+SB2,-SB5,Z345,DR35,SR4,%
     CRZ35,SUB1,ONE,OFF1,SB258,ZO,Z28]
    [C,N,C,O,O,
     CH,COOC_2H_5,COOC_2H_5]
\stopchemical
```

In this case the offset is 1, which means that we use one extra character. Though definitions appear to be quite complicated at first, they can best be constructed by defining the separate parts first. The rotation can best be made last.



A new command is seen in the above example: CRZ. This command can be used to position an atom or a molecule in line with a bond, which is desirable in this case. This positioning could have been achieved with the command RZ, since one can influence the spacing with the second set: {\,O} instead of {O}.

OFF	Offset	CRZ	Centered Radical Atom

Table 10: Special commands

11 Text formulas

Along with the typesetting of structure formulas, the typesetting of reaction equations is supported too. Therefore, the previously described command \chemical has another two versions:

```
  \chemical{formula}
  \chemical{formula}{text}
```

This command will fit to the position in the text. This means that a difference is made between:

- a text-mode
- a mathematical text-mode
- a mathematical display-mode

If the command is given in the current text, automatically $ $ are set round the command. In this way the command \chemical{NH_4^+} delivers the formula NH_4^+.

The same result is achieved by placing the command between $ $. Therefore in both cases the second argument is omitted. If we type the command between $$ $$, the second argument is obliged. The second argument may be empty:

```
$$
  \chemical{2H_2}{} \chemical{PLUS}{}
  \chemical{O_2}{} \chemical{GIVES}{}
  \chemical{2H_2O}{}
$$
```

gives:

$$2H_2 \quad + \quad O_2 \quad \longrightarrow \quad 2H_2O$$

This formula can also be defined in a shorter way:

```
$$\chemical{2H_2,PLUS,O_2,GIVES,2H_2O}{}$$
```

or even:

```
$$\chemical{2H_2,+,O_2,->,2H_2O}{}$$
```

The more experienced TeXuser will notice that the plus-sign as well as the arrow are located at the baseline; i.e., just compare + and +. Independent of the size of reflection, the + and the ⟶ are aligned.

In addition to `PLUS` and `GIVES`, `EQUILIBRIUM` can be given, which delivers double arrows (`<->`).

This formula can also be typeset in the current text, where a smaller layout is chosen: $2H_2 + O_2 \longrightarrow 2H_2O$. It is also possible to show bonds. For example, the rather long sequence

```
  \chemical{H,SINGLE,CH,DOUBLE,HC,SINGLE,H}
```

gives $H — CH = HC — H$.
This can also be achieved in a shorter way:
`\chemical{H,-,CH,--,HC,-,H}`.
A triple bond can be called with `TRIPLE` or `--`: $HC \equiv CH$.

Now we just return to the display mode. The second argument can be used to give an explanation of the formula:

```
$$
 \chemical{2H_2}{hydrogen} \chemical{PLUS}{}
 \chemical{O_2}{oxygen}
 \chemical{GIVES}{violent} \chemical{2H_2O}{}
$$
```

gives:

$$\underset{hydrogen}{2H_2} \qquad + \qquad \underset{oxygen}{O_2} \qquad \xrightarrow{violent} \qquad 2H_2O$$

The size of the formulas that are placed in the text can be set up with the setup-command:

variable	values	default
textsize	small medium big	big

Table 11: Setups for text formulas

When `big`, `medium` or `small` is used,

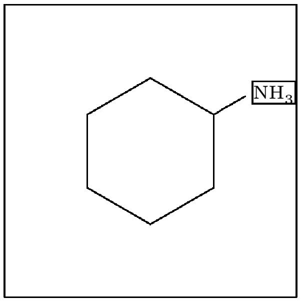
```
   \chemical{H,SINGLE,CH,DOUBLE,HC,SINGLE,H}
```

gives the successive formulas:
$H — CH = HC — H$ $H — CH = HC — H$ $H — CH = HC — H$

12 Subscripts

Sub- and superscripts are placed a little lower, as is recommended by Knuth in the TeXbook. As such, the example on page 179 in the TeXbook (that is chemically a bit strange) can be typeset by typing the command `\chemical{Fe_2^{+2}Cr_2O_4}`. This command gives $Fe_2^{+2}Cr_2O_4$. Without a correction, the command would have given the result: $\mathrm{Fe_2^{+2}Cr_2O_4}$.

The location of the subscript is by default determined by the `offset`: `HIGH` or `LOW`. The `offset` can be overruled by the commands of the same name.



Example 21

```
\startchemical[border=on,option=test,%
               alternative=2]
  \chemical[SIX,B,R1,HIGH,RZ1][NH_3]
\stopchemical
```

Although the location of the subscript can be set up by substituent, it is advisable to do this with the command `\setupchemical`. By doing so, all the formulas will be set up in the same way.

The keywords `LOW` and `HIGH` can also be used in text formulas, although this may lead to inconsistencies in layout. We can see that `\chemical{HIGH,H_2O}{}` results in H_2O and `\chemical{LOW,H_2O}{}` results in H_2O.

13 Installation

The package PPCHTeX is developed as an extension to ConTeXt but can be used with other packages. The macros can be found in the file `m-chemie.tex`, with the `m` standing for module. Therefore the macros are not standardly available.

Furthermore, the package can be used in combination with LaTeX. In this case the file `m-chemie.sty` is used too.[3] PPCHTeX can be activated by means of `\documentstyle`:

```
  \documentstyle[m-chemic]{}
```

In addition to the English interface that was described earlier, a Dutch interface is available. The Dutch version is loaded with:

```
  \documentstyle[m-chemie]{}
```

14 Some Dutch

The original interface of PPCHTeX is Dutch. One can switch back to Dutch with:

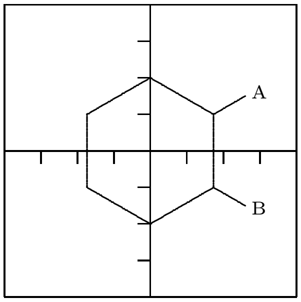
```
  \resetinterface
```

Switching back to English can be done with:

```
  \setinterface[english]
```

[3] The Dutch word chemie stands for chemistry in English.

Although more interfaces are possible, only the Dutch and English are implemented. In the file `m-chemie.sty` the interfaces are defined.



Example 22

```
\startchemie[assenstelsel=aan,kader=aan]
  \chemie[SIX,B,R12,RZ12][A,B]
\stopchemie
```

15 Extensions

All TeX-users are allowed to use the PPCHTeX-macros in another, non-commercial, way. However, some caution has to be taken into account, since the macros are still being developed, optimized and made more robust. Some macros may seem quite complicated, but appearances are deceptive. For example, commands in the shape of `\setup...` utilize the macros that support nesting and different ASCII-layouts. Compare for instance:

```
   \setupchemical[size=small]
```

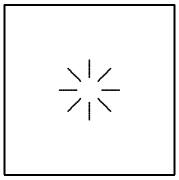
with:

```
   \setupchemical
     [size=small,
      scale=500,
      textsize=big]
```

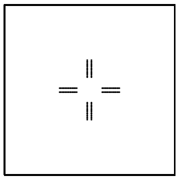
Setups can be given in an arbitrary sequence. When possible, spaces and newline-characters are suppressed and errors are announced.

The following overview shows the available structures. We use a somewhat smaller corps to save space. The manual of PPCHTeX *is a bit more extensive in its overviews.*

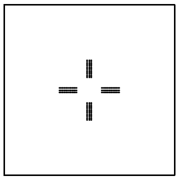
16 One



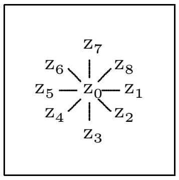
SB



DB

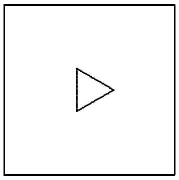


TB

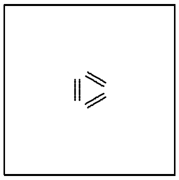


Z

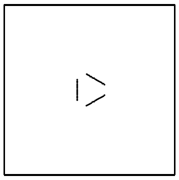
17 Three



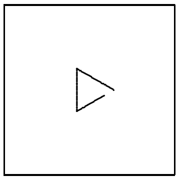
B



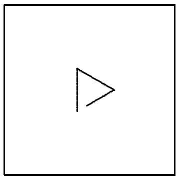
DB



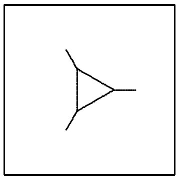
SB



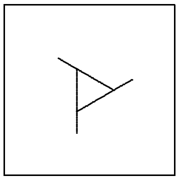
-SB



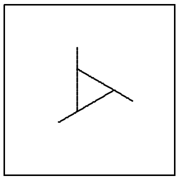
+SB



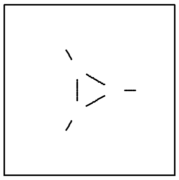
R



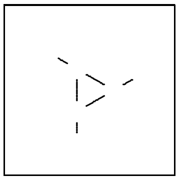
-R



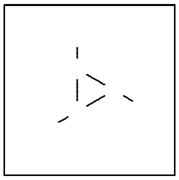
+R



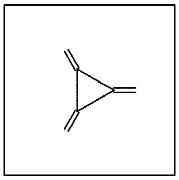
SR



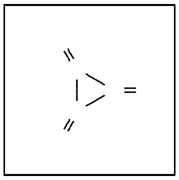
-SR



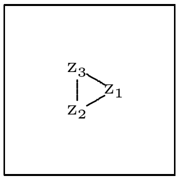
+SR



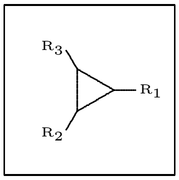
ER



DR

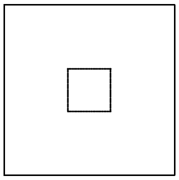


Z

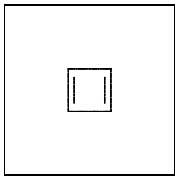


RZ

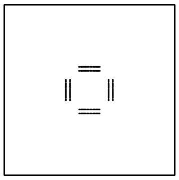
18 Four



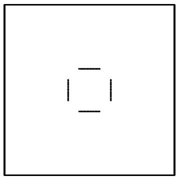
B

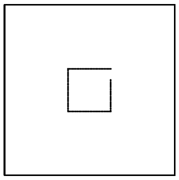


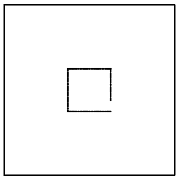
EB

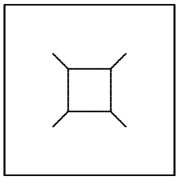


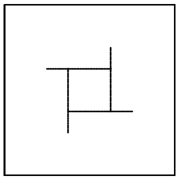
DB

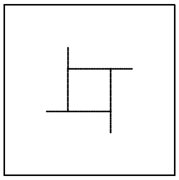

SB

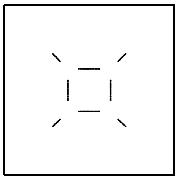

-SB

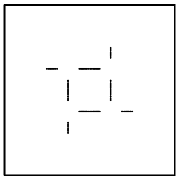

+SB

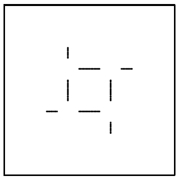

R

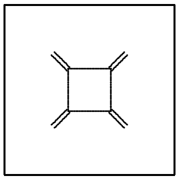

-R

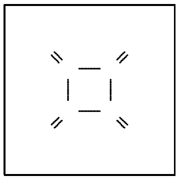

+R

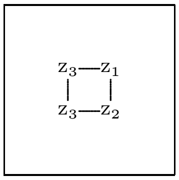

SR


-SR

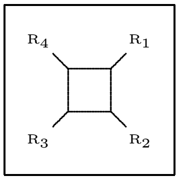

+SR


ER

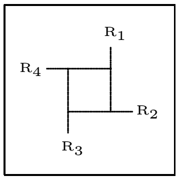

DR



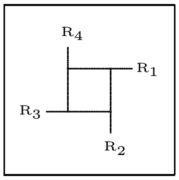
Z



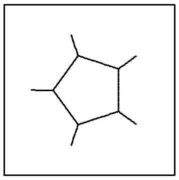
RZ

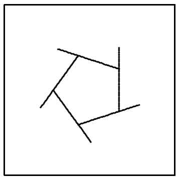


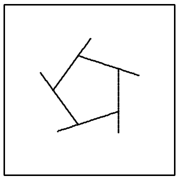
-RZ

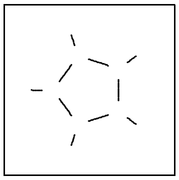


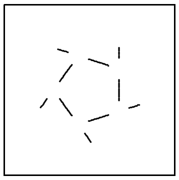
+RZ

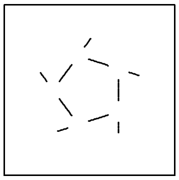

R

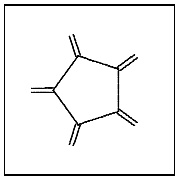

-R

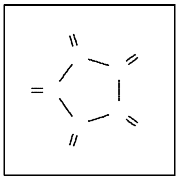

+R

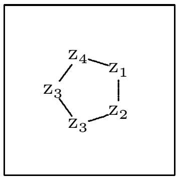

SR


-SR

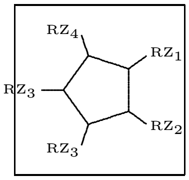

+SR


ER

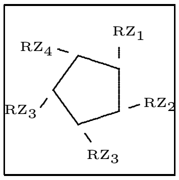

DR



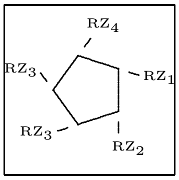
Z



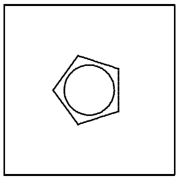
RZ

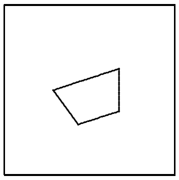


-RZ

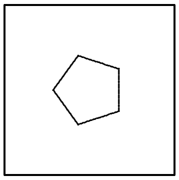


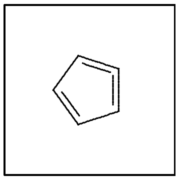
+RZ

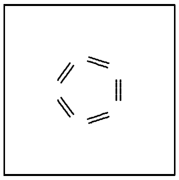

C

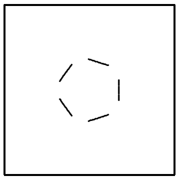

S

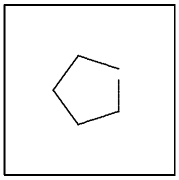
19 Five

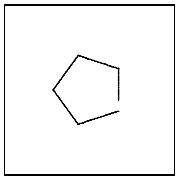

B


EB

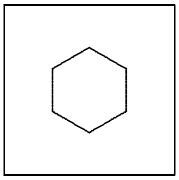

DB

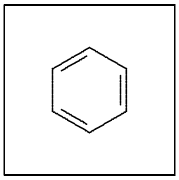

SB

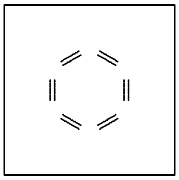

-SB

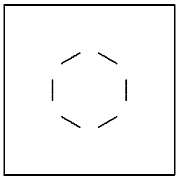

+SB

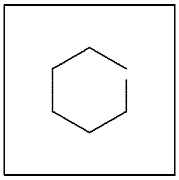
20 Six

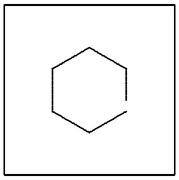

B

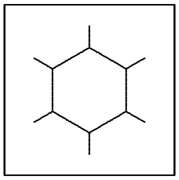

EB


DB

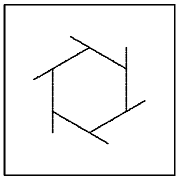

SB


-SB

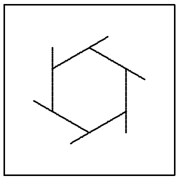

+SB



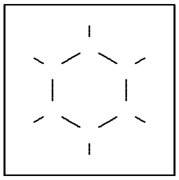
R



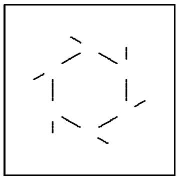
-R



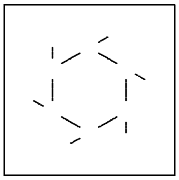
+R



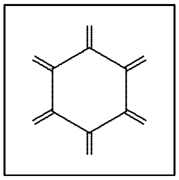
R



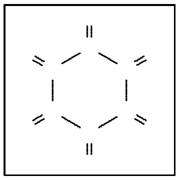
-R



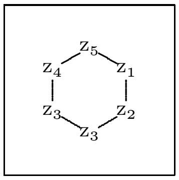
+R



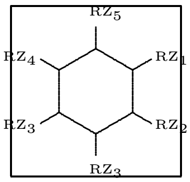
ER



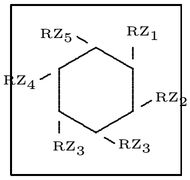
DR



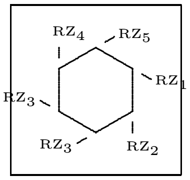
Z



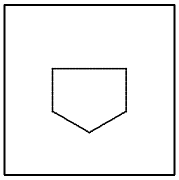
RZ



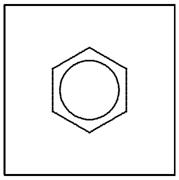
-RZ



+RZ

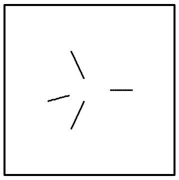


S

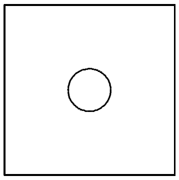


C

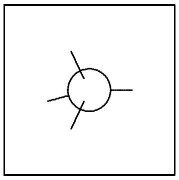
21 Carbon



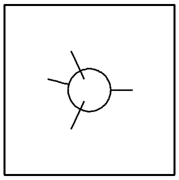
B



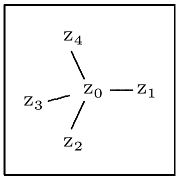
C



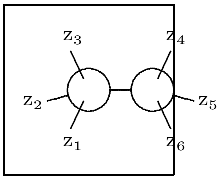
CB



MIR

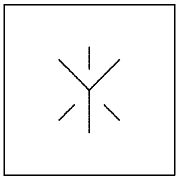


Z

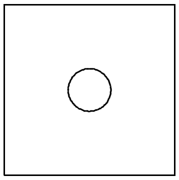


CB3

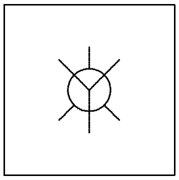
22 Newman



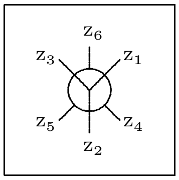
B



C

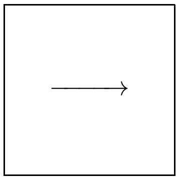


CB

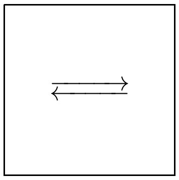


Z

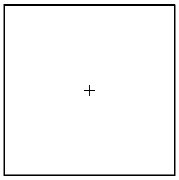
23 Symbol



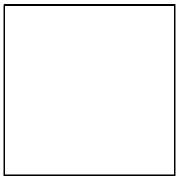
GIVES



EQUILIBRIUM



PLUS



SPACE

24 Other features

The following features of PPCHTeX are not described here, but one can find more on them in the manual.

- coloring parts of formulas
- active formulas in interactive texts
- automatic adapting to the corps size
- adapting to the resolution of printers
- the multilingual interface
- manipulating super- en subscripts

Also, the manual is a bit more extensive in its overview of the rotations and displacements.

◇ J. Hagen and A.F. Otten
Pragma
P.O.Box 125
8000 AC Zwolle
The Netherlands

LaTeX

A LaTeX Tour, part 1: The basic distribution

David Carlisle

Introduction

In this article[1] I hope to give a 'guided tour' around the files that make up the basic LaTeX distribution. Subsequent articles in this mini-series will cover other packages by the LaTeX development team, and also some of the main contributed packages.

The primary source for LaTeX is the 'CTAN[2]' network of archives, so if I refer to path names of files this relates to the CTAN file structure. Note however that if you obtained LaTeX as part of a 'prepackaged' TeX distribution, then these files may have been moved (typically documentation files may be separated from TeX source files). Hopefully this will not cause any confusion.

The Components of LaTeX

The LaTeX distribution at the CTAN archives is organised into the following directories.

`base` Contains the core LaTeX files. You need at least these files for a minimal LaTeX installation.

`unpacked` Includes *all* the files in `base` together with the result of 'unpacking' the source files. (More about this later.) Thus when obtaining LaTeX you should get either `base` or `unpacked`, but not both. Getting the former saves on time transferring the files, but getting the latter saves time that would be taken unpacking the source files, so which is preferable depends on the relative speed (and cost) of your machine and your connection to the archives.

`packages` Consists of seven independent LaTeX 'extensions' that are written and supported by the LaTeX developers (or the American Mathematical Society) — `amsfonts`, `amslatex`, `babel`, `graphics`, `mfnfss`, `psnfss` and `tools`.

These packages will be described in more detail later in the series.

`fonts` The metafont sources and TeX font metric files of a few fonts that LaTeX requires that are not part of the original plain TeX distribution.

`doc` This directory is not part of the main LaTeX distribution, it is generated by the CTAN archives. As a convenience for those people that have not yet installed LaTeX, some of the main introductory documentation files which are available as LaTeX files in the base distribution are made available in this directory as dvi and PostScript files.

`contrib` This directory contains an ever growing number of contributed LaTeX packages, and other extensions, that have been contributed by LaTeX users. They are not part of the 'official' LaTeX distribution, but many of them form a vital part of any 'working' LaTeX installation. The packages are divided into two subdirectories '`supported`' and '`other`', however at the current time one should ignore this distinction when looking for packages to fetch from the archives. Contrary to expectations some of the best supported packages are distributed (at their author's request) from `contrib/other`.

Unfortunately (for mainly historical reasons) people in search of contributed LaTeX packages also need to look in more distant CTAN directories. Firstly, the `macros/latex209/contrib` area on CTAN contains packages that were written for the previous version of LaTeX that has been obsolete since June 1994. Any packages that are still distributed from this LaTeX 2.09 tree are likely to be less than well supported, but you can still find some useful files there. Secondly, there are some packages that work with multiple formats, not just LaTeX, and these are to be found under `macros/generic` or in their own top-level directory, such as `macros/musictex`.

Documentation in the Base Distribution

The documentation that comes with LaTeX is of two forms: plain (ASCII) text files with extension `.txt`, or LaTeX documents distributed as LaTeX source with extension `.tex`. Generally speaking the text files are mainly of interest to people installing LaTeX, who may need information before they have a working system. Information of more general interest to LaTeX users is normally distributed as a LaTeX document.

[1] This is a slightly edited version of an article that first appeared in the UKTUG journal *Baskerville* 6.1.

[2] The Comprehensive TeX Archive Network: `ftp.tex.ac.uk` (UK), `ftp.dante.de` (Germany), in the USA the mirror `ftp.cdrom.com` of the UK archive should be used in preference to the original CTAN host of `ftp.shsu.edu` which is unfortunately unreliable at the present time.

The ASCII text files

Installation instructions

`00readme` Provides a general introduction to the system, and should be the first file to look at when installing LaTeX for the first time.

`install` Provides 'generic' installation instructions, but for many TeX versions more specific instructions that have been contributed by the authors or users of those systems; thus `emtex` gives instructions for the popular emTeX implementation, `web2ctex` gives specific instructions on installing under UNIX, etc.

`tex2` If you have a TeX that pre-dates version 3.0 (which was released in 1989) by far the best thing to do is to update your TeX, but if that is really not possible this file details how LaTeX may be built under TeX2.

`autoload` Describes the installation of an 'autoloading' version of LaTeX. This produces a much smaller format by saving less common commands in external files rather than in memory. These files are automatically 'autoloaded' as required. This version of LaTeX is particularly recommended if you are using a small installation (for instance a 'small' emTeX on a sub-386 PC).

Other text files

`legal` Contains the copyright notices and distribution conditions for LaTeX.

`bugs` Contains instructions on how to compile a bug report (see below).

`patches` Describes the LaTeX patch mechanism that is used for distributing small updates between the 'full' releases. This file also contains a list of all the files that have changed since the last full release.

`changes` A Change Log of all the changes made to the LaTeX files. This is mainly intended for internal use by the LaTeX developers, but some people like to read it.

The LaTeX 'guides'

These documents are distributed as LaTeX source (i.e., `.tex` files) although as noted in the above introduction, the CTAN archives distribute most of them in ready-formatted versions in the directory `latex/doc` so you can read these before installing LaTeX if you wish. Unlike the ASCII text files described above, most of these documents are primarily intended for *users* of the system rather than system managers and software installers.

`usrguide` *LaTeX 2ε for Authors.* This document describes all the main new features of the 2e release of LaTeX. It was written originally with the user of the old LaTeX 2.09 in mind, but newcomers to LaTeX who have never used the old version should still gain something by reading this document. It does not however cover the majority of LaTeX commands that were not changed, and so it is not a substitute for a full LaTeX manual.

`clsguide` *LaTeX 2ε for class and package writers.* A companion to to `usrguide`, gives details of the LaTeX commands for structuring class files and extension packages.

`fntguide` *LaTeX 2ε font selection.* For font addicts only, but if you want to know the detailed specification of the 'New Font Selection Scheme' commands, here is the place to look.

`cfgguide` *Configuration options for LaTeX 2ε.* Discusses what you can (and can not) do to configure a LaTeX installation to the requirements of your local site.

`ltx3info` *The LaTeX3 Project.* A brief summary of the aims of the LaTeX3 project, the group of volunteers that has taken on the maintenance and development of LaTeX.

`modguide` *Modifying LaTeX.* This document discusses some of the rationale behind the LaTeX distribution conditions as expressed in `legal.txt` and `cfgguide.tex`. Unless you are making a distribution of a modified version of LaTeX, or are particularly interested in software copyright issues, you probably do not want to read this.

LaTeX News

As well as these larger documents there are a series of one-page 'newsletters'. A new one is produced with each full release of LaTeX. These detail any changes that have occurred in LaTeX or the main extension packages over the six months since the previous release. (LaTeX releases occur at regular intervals, in June and December of each year.) Currently the four files `ltnews01`–`ltnews04` are distributed; these correspond to the four releases of LaTeX since June 1994.

Example Documents

There are two (very) small example documents, as described in the LaTeX book by Leslie Lamport.

`small2e` A very small (1 page) LaTeX document.

`sample2e` A slightly larger document.

Documented sources

The source for the LaTeX format, and for all the packages and classes in the core distribution is distributed as '`dtx`' files. These are LaTeX documents which may be processed in the usual way to produce typeset documentation. For example a command such as `latex ltpictur.dtx` would produce documented source of the picture mode commands. The files with names of the form '`lt....dtx`' make up the source of the LaTeX format. If you want to produce a combined document incorporating all these files, you may process `source2e.tex`. This document will produce a typeset version of the LaTeX sources, together with change log and index. It is well over 500 pages long, and so may take a long time to produce. It may produce an index that is too large to be handled by the 'makeindex' program on smaller machines.

Errata

The principal documentation for LaTeX is the two books *LaTeX: A Document Preparation System*, and *The LaTeX Companion*. Errata for these (and the German edition of *The Companion*) are available as `manual.err`, `compan.err` and `begleit.err`.

The LaTeX Bug Report Database

As described in the file `bugs.txt` mentioned above, the LaTeX3 project maintain a database of bug reports for LaTeX.

If, after checking with colleagues, reading the manual, etc., you decide that some behaviour of LaTeX is incorrect then you may send a message to the LaTeX bug database. Before doing this you should check that your LaTeX is not more than one year old (the bug may have been fixed in a recent release). If you have access to the World Wide Web, you may access the database and see if the problem is already reported by using the search page accessible from: `http://www.tex.ac.uk/CTAN/latex/bugs.html`.

If you decide to send a report, two files are available to help compose a message in the correct format:

`latexbug.tex` LaTeX this file and you will be prompted for information such as your name, and the name of a test file that shows the problem. A mail message will be written to the file `latexbug.msg` which should be sent to `latex-bugs@uni-mainz.de`. (You should *always* use `latexbug.tex` to generate messages to be sent to this bug address. It is an interface to a database (the GNU GNATS problem tracking system) and can not handle messages that are not in the special format written by `latexbug.tex`.

`latexbug.el` For users of the GNU Emacs text editor, a more convenient interface is provided by this file. It runs `latexbug.tex` automatically, and provides online help for filling in the various fields, and finally automatically mails the message to the correct address.

Docstrip files

As mentioned above, LaTeX is distributed as documented sources. The files that are actually used by TeX are extracted from these files by running `docstrip.tex`. The LaTeX distribution contains many files with extension `.ins` that control how `docstrip` extracts each file. Most of these are never used individually, as they would just 'unpack' one small part of the distribution. The file `unpack.ins` is a 'master' installation script that calls the smaller install files in turn and so unpacks the whole distribution. Normally running TeX on this file is the first step in installing LaTeX. This step may be omitted however if the `unpacked` directory is obtained from CTAN rather than `base`. `unpacked` is *exactly* the result of obtaining `base` and running TeX on `unpack.ins`. If you have a slow machine you may prefer this route as it saves unpacking time, but conversely it requires downloading more files, so if you are transferring the files via a slow connection such as a modem then you may prefer to get the smaller '`base`' distribution.

There are three install files that are *not* included into `unpack.ins` so you may have need to run these if you need the following features.

`autoload` Processing `autoload.ins` will generate the source file for the 'autoload' version of LaTeX, `latexa.ltx`, as described in the file `autoload.txt`. This should be processed with iniTeX to create a format file to be used in place of the standard `latex.fmt`. As well as the modified format, various packages are created containing the code that has been taken out of the format. Normally these do not need to be invoked explicitly as they are loaded on demand when they are needed. Currently the following package files are produced.

`autopict` Source for `picture` mode.

`autotabg` Source for `tabbing` environment.

`autoerr` The texts of most LaTeX error commands.

`autofss1` Less used font selection commands.

`autoout1` Code related to `\enlargethispage`. The autoload format is still quite experimental, and so the range of such 'autoloading' packages may change with future releases.

`cmextra` Processing `cmextra.ins` installs the '`fd`' files for the 'concrete' variants of the Computer Modern fonts, and also the AMS Cyrillic fonts.

`olddc` If using the Computer Modern fonts in the 8-bit 'T1' encoding, LaTeX defaults to using the 'dc fonts'. During 1995 these fonts were updated and the names of the fonts *changed*. Thus the 10 pt roman font corresponding to `cmr10` is now `dcr1000` rather than `dcr10`. The install file `unpack.ins` includes `newdc.ins` so by default LaTeX will use the new 1995 names (dc fonts release 1.2 or later) when using T1 font encoding. If you still have the old dc fonts, then you must process `olddc.ins` to produce suitable `fd` files referring to the old names.

The Standard LaTeX Classes

The general appearance of a LaTeX document, and the specification of the commands available is specified in a *document class*. This may be further modified by loading *packages*, as described in `usrguide`. In this section I give a brief overview of the available classes in the base distribution. They all have extension `.cls` (after being unpacked from the `.dtx` source file during the installation process).

`article` 'Article Class'. In some sense the canonical reference class against which all others are judged. This class (which is generated from the same `classes.dtx` source as report and book described below) is a mixed blessing. On one hand it provides quite a rich collection of commands for marking up documents that means that it serves well as as the basic 'generic' class to be used when no more suitable specific class is available. On the other hand the visual appearance of documents produced with this class is very distinctive. Many people who say they "don't like LaTeX" and so use some other format such as plain, in fact are misled into believing that LaTeX *is* this class. In fact by loading article and then making small adjustments one can produce very different visual designs. The class files for *Baskerville* and *TUGboat* are examples of such non-standard classes based on article.

However for many purposes, portability is more important than original typographical design, and in these cases the article class has the big advantage of being installed at all LaTeX sites.

`report` 'Report Class'. Very similar to article (and produced from the same source). The main differences are that this class has a higher level of sectioning command (`\chapter`) than is available in article, and the front matter is typeset differently.

`book` The book class is again very similar to report with the addition of a few extra features for controlling the front matter and back matter. It is unlikely that you would want to use this class 'as is' for a book, as you would almost certainly want to spend some effort (and perhaps money!) on an original design. However it can be used as a basis or example of the implementation of a LaTeX class for book production.

`letter` This provides commands for producing one or more letters. Many sites use this as a basis for producing site-specific letter class files, for instance with a modified heading that inserts a departmental logo and address.[3]

`proc` Proceedings class. This is a variant of article class (and inputs the `article.cls` file when used). It defaults to two column mode and makes one or two other small adjustments. It may be used as a model for how make a class that builds on another.

`slides` The slides class. This class essentially provides the functionality that was formally built into SLiTeX. It provides a mechanism for producing pages suitable for projecting on an overhead projector. It is described in the LaTeX book, and some people like it; however if you are making a lot of such presentations you may prefer to look at the contributed classes seminar (T. v. Zandt) or foiltex (J. Hafner). These provide alternatives to the standard class that many people find more useful.

As well as these 'Standard Classes' the base distribution contains a few other special purpose classes.

`minimal` This is the minimal LaTeX class. It just sets up a text area, and a font in a single size. None of the normal sectioning or font size commands are available. This class is not intended to be used in documents, but it is often useful when testing macros as it loads very quickly.

`ltxguide` A special purpose class for the 'LaTeX guides' mentioned earlier.

`ltnews` The class file used for the 'LaTeX News' news sheets.

[3] One should be able to find details of such local variants in the famous 'local guide'.

`ltxdoc` This class is used in all the `dtx` documentation files. It is based on the article class and the doc package, but with additional commands for documenting the LaTeX sources. It was not conceived as a class for general use, but some people find it convenient to use it when documenting their own package files.

Standard Packages

Encoding Packages

One of the main features of the 2e release of LaTeX is that it attempts to remove all 'hard wired' assumptions about the encodings being used, both for input and also in the fonts used for typesetting.

It maintains a strict distinction between the *Input Encoding* and the *Output Encoding*. The input encoding relates to the text that you type; this may be a standard encoding such as ASCII (the traditional 7-bit encoding) or ISO Latin-1, or a platform-specific encoding such as 'Windows ANSI' as used on MicroSoft Windows 3.x machines. The output encoding for text fonts is usually either OT1 (the encoding devised by Knuth and implemented in the original Computer Modern TeX fonts) or T1 (the new TeX encoding also known as 'Cork' after the meeting where it was agreed).

LaTeX maintains this separation by *always* translating input to an *Internal Encoding*. This is essentially traditional TeX 7-bit input. This internal encoding is then translated to the encoding used in the font without reference to the original input mechanism used. Thus if you specify an input encoding that includes the character é you may type that directly at the keyboard, and see it as a single character; however internally LaTeX will treat this as `\'{e}`. If you are using 7-bit OT1 encoded fonts this command will use the `\accent` primitive to add an acute to the e; however if you are using T1 fonts, the existing é will be accessed directly. Note however that the position of é in the output encoding (T1) is typically *different* from the position of the character in the input encoding used.

`inputenc` Specifies that an 8-bit input encoding is being used. A package option should always be used which sets up the default encoding. The currently available options include latin1, latin2, ansinew, cp437, cp437de, applemac. (The two IBM codepage 437 variants differ in just one slot; the former uses β, the latter uses ß.)

So typical usage (to specify ISO Latin-1 input conventions) would be:
`\usepackage[latin1]{inputenc}`

`fontenc` Specifies the default output encoding for text fonts. Currently the available options are OT1 and T1. So to specify that fonts in the the T1 (Cork) encoding be used in the document one would declare:
`\usepackage[T1]{fontenc}`

Remaining Packages in the Base Distribution

`alltt` Defines the alltt environment, similar to verbatim except that `\`, `{` and `}` retain their usual TeX meanings.

`doc` The package defining the commands used for documenting all the LaTeX code in the distribution.

`shortvrb` This package (really a small part of the doc package) defines the `\MakeShortVerb` command that allows shorthands like `|\foo|` instead of `\verb|\foo|` This is very convenient if you are documenting TeX or some other situation where you need to make a lot of use of short sections of verbatim text.

`exscale` For mainly historical reasons LaTeX always uses the math extension font (used for brackets and sum and integral signs etc.) at the same size, whatever the current font size. This package modifies this behaviour so that magnified fonts are used at larger sizes. At the same time it makes the plain TeX commands `\big`, `\bigg` etc., work as expected in conjunction with LaTeX size commands.

`flafter` LaTeX floats such as the figure and table environment can float *up* to the top of the current page. This means that it is possible that the figure appears before its first reference. Some publisher's styles do not allow this. flafter redefines the float placement algorithm so that a float never appears before its position in the source file, so by using this package, and placing the figure environment after the first reference to the figure, one can ensure that a figure will appear after the reference.

`graphpap` The `\graphpaper` command produces a grid for use in the picture environment.

`ifthen` Provides an 'if ... then ... else ...' programming construct for use in LaTeX packages. Many of the examples in *The LaTeX Companion* assume this package has been loaded.

`makeindx` Implements support for generating an index.

`pict2e` This package produces an error message to say that it has not been written. The documentation in the `pict2e.dtx` source file suggests alternative packages for extending picture mode, pspicture (D. P. Carlisle), pspic (K. K. Thorup), and the very powerful pstricks (T. V. Zandt).

`showidx` This causes the argument of each `\index` command to be printed on the page where it occurs. See also `idx.tex` described below.

`syntonly` Used to process a document without typesetting it. On some systems this speeds things up considerably, and so may (possibly) be useful while debugging documents.

`tracefnt` This allows you to control how much information about LaTeX's font loading is displayed.

`latexsym` Loads the special LaTeX symbol font and then defines commands such as `\Box` that use this font. These commands were defined by default in LaTeX 2.09.

`newlfont` Defines 'old' font commands to act in the 'new' way. For example it makes `\rm` essentially equivalent to `\rmfamily`. This package is not now recommended but is distributed so old documents written using the LaTeX 2.09 version of this package still work.

`oldlfont` A companion to newlfont. This package is only to be used for old documents that used the LaTeX 2.09 package of the same name.

Font Definition Files

Unpacking the LaTeX distribution creates dozens of 'font definition files' with extension '`.fd`' from their documented sources (with extension `.fdd`). These map the internal LaTeX model of fonts on to the external file names as used on your system. Normally you never need to load these explicitly into a LaTeX document and they will not be considered in detail here except to say that if you obtain some new fonts from the TeX archives, make sure to also get the related `fd` files, and install them where LaTeX can find them.

Makeindex Styles

The distribution includes three styles (with extension `.ist`) for the *makeindex* index generator. They modify the makeindex defaults so as to work with the special requirements of the doc package.

`gind` Produces indices of command definition and use.

`gglo` Produces 'change log' entries (using the LaTeX `\glossary` command rather than `\index`).

`source2e` This style is produced only if the LaTeX document `source2e.tex` is processed. It is almost identical to `gind.ist` but defines 'I' to be in the series 'I–J–K' rather than 'I–II–III'. This is needed for the numbering conventions used in that document.

Miscellaneous Utilities and Files

`idx.tex` Print out index entries in your document.

`lablst.tex` Generate list of labels used in a document. You may prefer instead to have the labels show up in the margins of your drafts, in which case use the showkeys package from the 'tools' collection to be described later in this 'tour'.

`ltxcheck.tex` This 'document' should always be processed after LaTeX has been installed. It produces no output but checks that various components of the system are configured correctly for your machine type.

`nfssfont.tex` Test file for testing a font. A more extensive font test is available if you use the fontsmpl package from the 'tools' collection.

`testpage.tex` Test file for checking the accuracy of a printer. This is particularly useful to see if you need to specify any offsets to your printer driver to ensure that the printed text is correctly positioned on the paper.

`Makefile.unx` A very simplistic template 'Makefile' for installing the LaTeX base distribution under UNIX. Many UNIX TeX distributions come with far more suitable installation procedures. For example the excellent 'teTeX' distribution allows you to install TeX, LaTeX, metafont, dvips, xdvi, and a host of other utilities and fonts just by typing `sh install.sh`.

`latex209.def` This file is loaded whenever a document beginning with `\documentstyle` is seen. It forces LaTeX into '2.09 compatibility mode' which is a slow, but a fairly accurate, emulation of the old version of LaTeX. This enables old documents to be processed under the current system.

Conclusion

The base distribution described here is only really a bare minimum LaTeX installation. Most sites would want to install all of the LaTeX files from the CTAN `latex/packages` directory, and some of the contributed packages as well. These other files will be covered in future installments of this 'tour'. Part 2 will cover the tools and graphics distributions. Part 3 will cover babel, mfnfss and psnfss. Part 3 will cover the AMS LaTeX distributions amsfonts and amslatex. If I am still feeling keen, later parts may venture out to describe some of the more popular contributed packages.

◇ David Carlisle
Mathematics Department
Manchester University
Oxford Road
Manchester, England M13 9PL UK
carlisle@ma.man.ac.uk

Abstracts

Les Cahiers GUTenberg Contents of Recent Issues

Numéro 20 — mai 1995

Editor's note: Thematic issue entitled "Character encodings from ASCII to Unicode." The issue contains 102 pages in all, with 3 very detailed articles: two coming out of the very successful conference to launch Omega in Geneva (16 March 1995), and one a modified version of a paper presented by Bigelow and Holmes at the RIDT'94 Conference (Darmstadt, April 1994).

In memoriam : Cathy Booth; pp. i–ii

"This issue of *Cahier GUTenberg* is dedicated to Cathy Booth.

Cathy will be remembered with affection by many TEX enthusiasts around the world as she regularly attended TEX meetings in both Europe and North America. For those fortunate enough to have been there, memories of the hugely successful TEX88 conference at Exeter University, UK, will be a lasting memorial to her infectious vivacity and organisational talent. She was a founder-member of the UK TEX Users' Group, becoming a very active committee member, and also a gifted teacher over a wide area of computer-controlled design and typesetting."

Jacques André, Michel Goossens, Yannis Haralambous, Éditorial : Multilinguisme et codage [Editorial: Multilingualism and coding]; pp. iii–iv

The joint editorial begins by reminding readers that while GUTenberg's mission does not focus exclusively on TEX, it is the best solution for technical and document exchange, compared with other more "modern", more commercial products. The editorial then moves on to outline plans for future issues of the *Cahiers*, exploring the general theme of electronic document exchange; the previous issue, no. 19, had been the first in the series, looking at the Web.[1] Subsequent issues (in addition to the current one) include the following:

23 Omega — 32-bit multilingual TEX
24 Ligatures — including reference to non-Latin based alphabets
25 TEX and Arabic — proceedings of a conference at INALCO[2]
26 Courier — in praise of typewriter-style monospaced fonts and their role in typography world-wide

Finally, the editorial briefly introduces each of the three essay-length articles, including a note that the Bigelow and Holmes article was deemed important because it is about the first commercial application of Unicode, and thus is another facet of the many issues raised during the course of the Omega Conference held at CERN in March 1995.

Jacques André, Michel Goossens, Codage des caractères et multi-linguisme : de l'ASCII à Unicode et ISO/IEC-10646 [Character coding and Multilingualism: From ASCII to Unicode and ISO/IEC-10646]; pp. 1–53

Author's abstract: "After reviewing the difference between glyphs and characters, we discuss character exchange standards, like ASCII and ISO-Latin 1. Then we turn our attention to Unicode, a 16-bit encoding standard that will eventually represent the characters of all living languages and thus will make it possible to exchange without problems texts written in the languages spoken in various parts of the world. ISO/IEC-10646 is a 4-byte generalisation — the first two bytes coinciding with Unicode — but whose full 32-bits wide encoding space allows the representation of special or ancient characters."

[1] See *TUGboat* 15, #4, pages 498–99.

[2] Institut National des Langues et Civilisations Orientales

Yannis Haralambous, John Plaice, Ω, une extension de TEX incluant Unicode et des filtres de type Lex [Ω, a TEX extension including Unicode and Lex-type filters]; pp. 55–79

Author's abstract: "Ω consists of a series of extensions to TEX that improve its multilingual capabilities. It allows multiple input and output character sets, and will allow any number of internal encodings. Finite-state automata can be defined, using a `flex`-like syntax, to pass from one coding to another.

In this paper both a technical introduction and a few applications of the current implementation of Ω are given. The applications concern typesetting problems that cannot be solved by TEX (consequently, by no other typesetting system known to the authors). They cover a wide range, going from calligraphic script fonts (Adobe *Poetica*), to plain Dutch/Portuguese/Turkish typesetting, to vowelized Arabic, fully diacriticized scholarly Greek, or decently kerned Khmer.

A few problems Ω cannot solve are mentioned, as challenges for future Ω versions."

Charles Bigelow, Kris Holmes, Création d'une police Unicode [Creating a Unicode font]; pp. 81–102

Author's abstract: "The international scope of computing, digital information exchange, and electronic publishing has created a need for worldwide character encoding standards. Unicode is a comprehensive standard designed to meet such a need. To be readable by humans, character codes require fonts that provide visual images — glyphs — corresponding to the codes. The design of a font developed to provide a portion of the Unicode standard is described and discussed."

[Editor's note: This article originated as a paper presented at both RIDT'94 and EP'94 in Darmstadt, April 1994. The paper was subsequently presented at the TUG'94 meeting in Santa Barbara, entitled "Lucida and TEX: lessons of logic and history", but not published in the proceedings, as it had already appeared elsewhere: "The Design of a Unicode Font", *Proceedings of RIDT'94: Electronic Publishing — Origination, Dissemination and Design*, 6(3), pages 289–306, 1993.]

(Compiled by Christina Thiele)

Calendar

1996

Mar 20 UK TeX Users Group, University of Warwick. TeX and the Internet. Local organizer, Malcolm Clark. For information, e-mail `uktug-enquiries@ftp.tex.ac.uk`

Mar 27 – 29 DANTE '96 and 14th general meeting of DANTE e.V., Universität Augsburg, Germany. For information, contact Gerhard Wilhelms (`dante96@Uni-Augsburg.de`).

Apr 2 DANTE TeX–Stammtisch at the Universität Bremen, Germany. For information, contact Martin Schröder (`MS@Dream.HB.North.de`; telephone 0421/628813). First Tuesday (if not a holiday), 18:00, Universität Bremen MZH, 28359 Bremen, 4th floor, across from the elevator.

Apr 4 DANTE TeX–Stammtisch at the Universität Karlsruhe, Germany. For information, contact Klaus Braune (`braune@rz.uni-karlsruhe.de`; telephone 0721/608-4031). First Thursday, 19:30, Rechenzentrum der Universität Karlsruhe, Zirkel 2, 3.0G Raum 316.

Apr 11 DANTE TeX–Stammtisch, Wuppertal, Germany. For information, contact Andreas Schrell (`Andreas_Schrell@FernUni-Hagen.DE`, telephone 0202/502354). Second Thursday, 19:30, Gaststätte Yol, Ernststraße 43, (near Robert-Daum-Platz), 42117 Wuppertal.

Apr 24 DANTE TeX–Stammtisch, Hamburg, Germany. For information, contact Volker Hüttenrauch (`Volker_Huettenrauch@HH.Maus.DE`. Last Wednesday, 18:00, details to be posted to the `tex-d-l` list.

May 2 – 4 BachoTeX '96: GUST 4th Annual Meeting in Bachotek, Poland, "The World around TeX". For information, contact Jola Szelatyńska (`mjsz@cc.uni.torun.pl`).

Status as of 11 March 1996

May 2 DANTE TeX–Stammtisch at the Universität Karlsruhe, Germany. (For details, see Apr 4.)

May 7 DANTE TeX–Stammtisch at the Universität Bremen, Germany. (For details, see Apr 2.)

May 9 DANTE TeX–Stammtisch, Wuppertal, Germany. (For details, see Apr 11.)

May 12 – 16 SGML Europe '96, Park Hotel, Munich, Germany. For information contact the Graphic Communications Association, `munich96@aol.com`, Fax: +1 703-548-2867.

May 24 – 26 Nordic TUG meeting, Tallin, Estonia. For information, contact Enn Saar (`saar@aai.ee`).

May 29 GUTenberg '96, on TeX distributions, and GUTenberg Annual Meeting, Nanterre, Paris, France. For information, contact the Association GUTenberg (`gut@irisa.fr`, phone/fax: (33 1) 30 87 06 25).

May 29 DANTE TeX–Stammtisch, Hamburg, Germany. (For details, see Apr 24.)

Jun 4 DANTE TeX–Stammtisch at the Universität Bremen, Germany. (For details, see Apr 2.)

Jun 10 **TUGboat 17 (2)** (proceedings issue): Deadline for receipt of revised articles, calendar items.

Jun 18 **TUGboat 17 (3),** (2nd regular issue): Deadline for receipt of *technical* manuscripts (tentative).

Jun 25 – 29 ALLC-ACH '96: Joint International Conference, Association for Literary and Linguistic Computing / Association for Computers and the Humanities, Bergen, Norway. For information, contact Espen Ore (`Espen.Ore@hd.uib.no`, Fax: +47 55 32 26 56) or visit `http://www.hd.uib.no/allc-ach96.html`.

Jun 26 DANTE TeX–Stammtisch, Hamburg, Germany. (For details, see Apr 24.)

Jul 2 DANTE TeX–Stammtisch at the Universität Bremen, Germany. (For details, see Apr 2.)

Jul 4 DANTE TeX–Stammtisch at the Universität Karlsruhe, Germany. (For details, see Apr 4.)

Jul 18 – 21 SHARP 1996: Society for the History of Authorship, Reading and Publishing, Fourth Annual Conference, Worcester, Massachusetts. For information, contact the American Antiquarian Society, `cfs@mark.mwa.org`.

Jul 28 – Aug 2 **TUG 17th Annual Meeting**: "Poly-TeX", Dubna, Russia. For information, send e-mail to `TUG96@pds.jinr.ru`. (See the Call for Papers, *TUGboat* **16** (4), p. 429.)

Jul 31 DANTE TeX–Stammtisch, Hamburg, Germany. (For details, see Apr 24.)

Aug 1 DANTE TeX–Stammtisch at the Universität Karlsruhe, Germany. (For details, see Apr 4.)

Aug 6 DANTE TeX–Stammtisch at the Universität Bremen, Germany. (For details, see Apr 2.)

Aug 20 **TUGboat 17 (4)**: Deadline for receipt of *technical* manuscripts (tentative). Theme issue, contributions by invitation.

Aug 28 DANTE TeX–Stammtisch, Hamburg, Germany. (For details, see Apr 24.)

Sep 3 **TUGboat 17 (3)** (2nd regular issue): Deadline for receipt of news items, reports (tentative).

Sep 3 DANTE TeX–Stammtisch at the Universität Bremen, Germany. (For details, see Apr 2.)

Sep 5 DANTE TeX–Stammtisch at the Universität Karlsruhe, Germany. (For details, see Apr 4.)

Sep 23 PODP, Workshop on Principles of Document Processing, Xerox Palo Alto Research Center, Palo Alto, California. For information, visit the PODP Web page at `http://www.cs.umbc.edu/conferences/podp/`.

Sep 24 – 26 EP96, the International Conference on Electronic Documents, Document Manipulation and Document Dissemination, Xerox Palo Alto Research Center, Palo Alto, California. *Deadline for submission of papers: 1 April 1996.* For information, contact `ep96@xsoft.xerox.com` or visit `http://www.xsoft.com/XSoft/ep96.html`.

Sep 25 DANTE TeX–Stammtisch, Hamburg, Germany. (For details, see Apr 24.)

Nov 12 **TUGboat 17 (4)**: Deadline for receipt of news items, reports (tentative).

Dec 8 – 12 SGML '96, Toronto, Canada. For information contact the Graphic Communications Association, Fax: +1 703-548-2867.

For additional information on the events listed above, contact the TUG office (415-982-8449, fax: 415-982-8559, e-mail: `tug@tug.org`) unless otherwise noted.

$$\Pi\mathrm{O}\Lambda\Upsilon\mathrm{T\!E\!X} \quad = \quad \left\{ \begin{array}{l} \text{Polytechnic} \\ \text{Polymath} \\ \text{Polyglot} \end{array} \right.$$

TUG'96

July 28 – August 2, 1996
Organization: TUG, *CyrTUG*, JINR

Come join us in late July in Dubna Russia, where TEX will have its first practical session in Cyrillic. For detailed information on the Conference site see the previous *TUGboat* **16**(4), pages 429–431).

Revised Deadlines

Copy of article to be reviewed	April 26
Reviewed article back to authors	May 13
Proposals for workshops, demos, poster session	May 20
Registration and transfer of the sum of $100 *per person* (non-refundable) to Dubna bank (see below)	May 31
Visa supporting information	June 5
Final text of paper	June 10
Start of Conference	*July 28*

Important Update

The *non-refundable* sum of $100 (USD only) per person should be transferred to the Moscow bank account (see below) before June 1st. Upon receipt of that sum an official invitation, necessary for obtaining a visa will be faxed to the participant — so be sure to include your FAX number. The remainder of the registration fee (between $450 and $500) will be payable in cash upon arrival at the Conference site in Dubna.

Name of Bank	Credit-Moscow Russia, Moscow, 113054 6-TH Monetchikovsky Per., 8
Swift Code:	CRMORUMM
A/C N:	00107091799

Visas

Most of the visitors from outside Russia will need a visa to attend the Conference. Therefore, for arranging a visa into Russia, participants should inform the Conference Secretariat (Mrs. N. Dokalenko, E-mail `nataly@ypr.jinr.dubna.su`; Fax 7 095 975 2381 or 7 09621 65 891) of their and (possibly) the accompanying person(s)'s full name, date of birth, citizenship, passport number, arrival and departure dates. The Secretariat will forward by fax the visa support message to the participants with which they should apply for visas to the nearest Russian Embassy or Consulate. Please note that you should apply for a visa valid for Dubna, Moscow, and Sergiev Posad[1].

Transportation

The Organizing Committee will arrange direct transportation by bus from Sheremetyevo-2 airport to Dubna (130 km north of Moscow). Not later than four working days before a participant wishes to be met at the airport should the Secretariat be informed of the flight number, precise date and time of arrival (Moscow time). It is our intention to have each participant met by a member of the Organizing Committee, who will accompany the Conference guests to Dubna.

Preliminary Program

– Saturday p.m. –

Arrival and registration

Dinner

– Sunday a.m. –

Late Registrations

Opening Ceremony

– Sunday p.m. –

Fonts

Karel Píška "Cyrillic Alphabets"

Jörg Knappen "DC and TC Fonts — A Status Report"

Rei Fukui (Presented by Knappen) "TIPA: A New IPA Font for LaTeX"

Olga G. Lapko "Full Cyrillic: How Many Languages?"

A.S. Berdnikov and O.A. Grineva "Computer Modern Typefaces as the Multiple Master Fonts"

A.S. Berdnikov and S.B. Turtia "VFComb 1.3 — A Program to Simplify Virtual Font Management"

[1] Those planning to visit other cities in Russia should obtain themselves the relevant documents and join them to the visa application, so that the needed names of the places to be visited can be entered on the visa form.

– Monday a.m. –

Encodings and Multilingual Support I

Yannis Haralambous and John Plaice "Omega and Cyrillic Alphabet Languages: a Synthesis of Available Resources"

Richard J. Kinch "Extending TEX for Unicode"

V.A. Konyshev, L.N. Znamenskaya, and S.V. Znamenskii "Various Russian Encoding Problems and a New Cyrillic Font Set"

Peter A. Ovchenkov "Cyrillic TEX Files: Interplatform Portability"

Alexander I. Rozhenko "On Cyrillic TEX and Multilingual Support"

– Monday p.m. –

TEX Systems I

John Plaice and Yannis Haralambous "Latest Developments in Omega"

Dag Langmyhr "StarTEX"

Gabriel Valiente Feruglio "Do Journals Honour LATEX Submissions?"

S.V. Znamenskii and D.E. Leinartas "A New Approach to TEX-related Programs with a User-friendly Interface"

17:00–18:30 TUG Business Meeting

– Tuesday a.m. –

TEX Systems II

Irina Makhovaya "TEX in Russia: ab ovo"

Joseph V. Romanovsky and Fedor L. Yeltsov "Strategies of LATEX Insertions"

Ivan G. Vsesvetsky "The Straight Gate to TEX"

Laurent Siebenmann "Electronic Posting and Archiving of TEX-composed Scientific Documents"

Strelkov S.A. and G.R. Epshtein "Making Indexes for VINITI's 'Mathematic' Abstracts Journal"

M. I. Grinchuk "How to Teach TEX to Repeat Signs when Breaking Formulae"

– Tuesday p.m. –

Buses depart to Sergiev Posad (packed lunch)

– Wednesday a.m. –

Graphics

Kees van der Laan "Turtle Graphics and TEX — a Child Can Do It"

Kees van der Laan "Graphics and TEX — a Reappraisal of METAFONT/MetaPost"

A.V. Astrelin "Graphic Algorithms: a Common-use Library"

A.S. Berdnikov, O.A. Grineva, and S.B. Turtia "Some Useful Macros to Extend LATEX's picture Environment"

– Wednesday p.m. –

Workshop on Electronic Documents

Michel Goossens "LATEX to/from HTML — an Update"

Sebastian Rahtz "LATEX, Hypertext and Acrobat"

Michel Goossens and Sebastian Rahtz "Overview of the Latest Developments in Electronic Publishing: HTML/SGML, DSSSL, Style Sheets, Amber, Java, ..."

Joachim Schrod "The TDS File System — A Short Introduction"

– Thursday a.m. –

Encodings and Multilingual Support II

M.M. Vinogradov "A User-friendly Multi-function TEX-interface Based on MultiEdit"

Yu. E. Hohlov, S.V. Klimenko, E.V. Pankratiev, and S.V. Znamenskii "The New 'Russian TEX' Project"

Olga Lapko and Irina Makhovaya "Questions of Babel and Cyrillic Compatibility"

A. Slepuhin "Methods of Multilingual Text Processing"

Yannis Haralambous "Extending Al-Amal to Unicode"

12:30 Formal Closing Ceremony of TUG '96

13:00 We will adjourn for an excursion on the Volga, with a farewell party

– Friday –

All Day Excursion to Moscow (packed lunch)

At the end of the day those not returning to Dubna to participate in courses can be left in the centre of Moscow or at the Airport.

– Saturday/Sunday –

To Be Announced

Courses or workshops for those interested.

Late-Breaking News

Production Notes

Mimi Burbank

Welcome to 1996! The Production Team, after working extremely hard to produce five issues of *TUGboat* in five months finally had a brief respite before jumping into the first issue for 1996. For this issue, we had to completely rebuild the executables in order to run Hagen's article (page 54). The parameter stack was bumped from 60 to 200.

Macros

Other than the production files (frontmatter and boilerplate), only one article was received in plain TeX, the remainder being in LaTeX 2$_\varepsilon$. Of the latter, only Hagen's file (page 54) was run in compatibility mode. All articles were received as fully tagged for *TUGboat*, using either `plain`-based or LaTeX conventions described in the Authors' Guide (see *TUGboat* 10, no. 3, pages 378 – 385).

The TeX installation at SCRI was updated in February of this year, and we are now using Thomas Esser's excellent *teTeX* packaging of Karl Berry's *web2c*. TUG is working with the UK and French user groups to publish a ready-to-run Unix CD-ROM of this setup in the very near future.

The LaTeX 2$_\varepsilon$ macro packages for *TUGboat* are in a constant state of evolution, and updated files and documentation will soon be available on the CTAN archives in `tex-archive/digests/tugboat`. I have been a reluctant convert (for over ten years) to the use of LaTeX, but the versatility of LaTeX 2$_\varepsilon$ has been well-demonstrated to me over the past six months.

Fonts

This issue has been set in Computer Modern fonts using Malyshev's BaKoMa PostScript Type 1 versions. Žubrinić's article (page 29) also uses Croatian fonts, and the author provided us with 35 `mf` files, three `pl` files and 21 `tfm` files.

Output

The final camera copy was prepared at SCRI on an IBM RS6000 running AIX, using the *Web2C* implementation of TeX.

Output was printed on a QMS 680 print system at 600 dpi.

Future Issues

The next *TUGboat* will be the Proceedings Issue for TUG'96. Issue 17(3) will be a regular issue, and will contain an article by Robin Fairbairns on "Macros for producing *TUGboat* articles under LaTeX 2$_\varepsilon$". Additionally, you can look forward to the Part 2 installments of Reckdahl's article on "Using EPS Graphics in LaTeX 2$_\varepsilon$ Documents", and Carlisle's "A LaTeX Tour", as well as highlights of the upcoming annual meeting which will be held in Dubna, Russia, in July of this year. Issue 17(4) will be guest-edited by Christina Thiele and Pierre MacKay. The theme will be "non-mathematical" TeX, TeX and the humanities, TeX and the world's languages, ...

Suggestions are welcome for prospective themes and guest editors. Send them to the Editor, Barbara Beeton (see address on page 3), or via electronic mail to `TUGboat@ams.org`.

BYLAWS of the TEX Users Group (“TUG”)

Article I
PURPOSES, POWERS AND NON-PROFIT STATUS

Section 1. Purposes. The TEX Users Group (“TUG”) has been formed exclusively for charitable, educational and scientific purposes, and specifically to identify, develop, operate, fund, support, promote and encourage charitable, educational and scientific programs and projects which will stimulate those who have an interest in systems for typesetting technical text and font design; to exchange information of same and associated use of computer peripheral equipment; to establish 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 connection with the foregoing and for any lawful purpose or purposes permitted under the Rhode Island Non-profit Corporation Act.

Section 2. Powers. TUG shall have the power, directly or indirectly, either alone or in conjunction or cooperation with others, to do any and all lawful acts and things and to engage in any and all lawful activities which may be necessary, or convenient to effect any or all of the purposes for which TUG is organized, and to aid or assist other organizations whose activities are such as to further accomplish, foster, or attain any of such purposes. The power of TUG shall include, but not be limited to, the acceptance of contributions in cash, in kind or otherwise from both the public and private sectors. Notwithstanding anything herein to the contrary, TUG shall exercise its powers only in furtherance of exempt purposes as such terms are defined in Section 501(c)(6) of the Internal Revenue Code of 1986 and the regulations from time to time promulgated thereunder.

Section 3. Non-Profit Status. TUG shall be non-profit and shall not have or issue shares of capital stock, and shall not declare or pay dividends. No part of the net income or profit of TUG shall inure to the benefit of any member, director, officer, or other individual, or to the benefit of any organization not qualified for tax exemption under Section 501(c)(6) of the Internal Revenue Code except as permitted by law. No substantial part of the activities of TUG shall be carrying on propaganda, or otherwise attempting to influence legislation (except as otherwise provided by Internal Revenue Code Section 501(h)), or participating in, or intervening in (including the publication or distribution of statements), any political campaign on behalf of any candidate for public office. Upon the dissolution of this organization, assets shall be distributed for one or more exempt purposes within the meaning of Section 501(c)(6) of the Internal Revenue Code or corresponding Section of any future Federal tax code, or shall be distributed to the Federal Government, or to a state or local government, for a public purpose.

Article II
OFFICES

TUG will have offices at such places both within and outside the State of Rhode Island as may from time to time be determined by the Board of Directors.

Article III
MEMBERS

Section 1. Constitution. The members of TUG will be such persons, natural or legal, who will meet such qualifications and requirements (including without limitation payment of initiation fees and dues) as from time to time may be established by the Board of Directors. The Board of Directors will be the sole judge of the qualifications of the members and its determination as to whether a person is or is not a member will be final. The Board of Directors may, in its discretion, create different classifications of members and prescribe different rights, privileges, qualifications or requirements for each class.

Section 2. Place of Meetings. All annual meetings of the members and all special meetings of the members called by the President or the Board of Directors will be held at such place, either within or outside the State of Rhode Island, as will be stated in the notice of meeting.

Section 3. Annual Meetings. Meetings of the members will be held in conjunction with TUG conferences. Such conferences will normally be held annually; otherwise, an annual meeting of the members will be held on the first Monday of August in each year if not a legal holiday in the place where it is to be held, and, if a legal holiday, then on the next day following which is not a legal holiday, beginning at 10:00 a.m. or at any other time designated in the notice of the meeting. At each annual meeting, the members will transact such business as may properly come before the meeting. In the event of the failure to hold said annual meeting at any time or for any cause, any and all business which might have been transacted at such meeting may be transacted at the next succeeding meeting, whether special or annual.

Section 4. Special Meetings. A special meeting of the members, for any purpose or purposes, may be called by the President or by the Board of Directors. Any such call will state the purpose or purposes of the proposed meeting.

Section 5. Notice of Meetings. Written notice of each annual or special meeting stating the place, day and hour of the meeting (and the purpose or purposes of any special meeting) will be given by or at the direction of the president, the secretary or the person or persons calling the meeting to each member entitled to vote at such meeting not less than ten nor more than sixty days before the meeting. Business transacted at any special meeting of members will be limited to the purposes stated in the notice of the meeting or any written waiver thereof.

Section 6. Quorum. Fifty (50) members present in person, will constitute a quorum at all meetings of the members. If, however, such quorum will not be present at any such meeting, the members entitled to vote thereat will have power to adjourn the meeting from time to time, without notice other than announcement at the meeting, until a quorum will be present. At such adjourned meeting at which a quorum will be present any business may be transacted which might have been transacted at the meeting as originally called. If adjournment is for more than thirty days, a notice of the adjourned meeting will be given to each member entitled to vote at the meeting. When a quorum is present at any meeting, the vote of the holders of a majority of the votes entitled to be cast and present in person will decide any question brought before such meeting, unless the vote of a greater number is required by law. A voice vote will normally be considered sufficient for business actions. A show of hands may be requested when the outcome is in doubt.

Section 7. Access to Documents. Nothing in these Bylaws shall be construed to limit the access of TUG members to TUG documents. Members requesting copies of any TUG document may be charged a reasonable copying fee and members requesting publications or mailing lists presented to the public for sale may be charged the same fee as the general public. Members requesting copies of documents to be used in performance of TUG related duties may request that the copying fee be waived. TUG documents include, but are not limited to: contracts, Board minutes, Executive Committee minutes, Finance Committee minutes, office procedure manuals, IRS filings, and written communications from or to the TUG office. This section does not authorize the release of any information that federal or state law protects from disclosure.

Article IV
DIRECTORS

Section 1. Powers. The affairs of TUG will be managed by the Board of Directors.

Section 2. Number. The number of directors will be not more than fifteen (15). Under very special circumstances, particularly deserving individuals may be designated as permanent Honorary Members of the Board, without vote, and without being included in the number of directors specified in this section.

Section 3. Composition. The Board of Directors will consist of the TUG President, Elected Directors, and Honorary Members.

Section 4. Honorary Members. The Grand Wizard, Donald E. Knuth, the Wizard of Fonts, Hermann Zapf, and the Founding Executive Director, Raymond Goucher, are designated as permanent Honorary Members of the Board.

Section 5. Nominations, Election and Term. Any member may be nominated for election to the Board by submitting a nomination petition in accordance with the TUG Election Procedures. Election of the directors shall be by written mail ballot of the entire membership, carried out in accordance with those same Procedures. Each director will hold office for a term of three (3) years. Directors may be re-elected for successive terms. Directors need not be residents of the State of Rhode Island.

Section 6. Meetings. The Board of Directors may hold meetings, both regular and special, either within or outside the State of Rhode Island. The first meeting of each newly elected Board of Directors will be held at such time and place as will be specified in a notice delivered as hereinafter provided for special meetings of the Board of Directors, or as will be specified in a written waiver signed by all of the directors. Regular meetings of the Board of Directors may be held without notice at such time and at such place as will from time to time be determined by the Board of Directors. Special meetings of the Board of Directors may be called by the President on two days' notice to each director, either personally or by mail or by telegram. Special meetings will be called by the President in like manner and on like notice on the written request of two directors. Meetings of the directors may be held by means of a telephone conference circuit and connection to such circuit will constitute presence at such meeting.

Section 7. Vacancies. Any vacancy occurring on the Board of Directors may be filled by the President. A director appointed to fill a vacancy will be appointed for the unexpired term of his or her predecessor in office. Any place on the Board to be filled by reason of an increase in the number of directors, or by a shortfall in the number of nominees at an election, may be filled by the President for a term of office continuing only until the next election of directors, provided that the number of Elected Directors does not fall below fifty (50%) percent of the number of directors specified in Section 2 of this Article.

Section 8. Quorum. At all meetings of the Board of Directors, fifty (50%) percent of Elected Directors will constitute a quorum for the transaction of business. The act of a majority of the directors present at a meeting at which a quorum is present will be the act of the Board of Directors, unless the act of a greater number is required by the Rhode Island Non-profit Corporation Act or by the articles of incorporation.

Section 9. Directors' Consent Vote. Any action required or permitted to be taken at a meeting of the Board of Directors or of any committee thereof may be taken without a meeting by instead taking a vote by mail, according to the following procedure. Any director may submit a motion in writing. Any other director may second the motion. Amendments to the motion are allowed, but not amendments to amendments. From the point that the motion on the floor has been seconded, there shall be a two-week period of discussion regarding the motion. After the discussion period, there shall be an active voting period of two weeks, after which voting shall be terminated. The votes shall then be tallied, counting the number of yeas, nays, and abstentions. At least fifty (50%) percent of Elected Directors must vote, otherwise the motion fails. When the fifty (50%) percent requirement is met, a motion shall pass when two-thirds of votes cast are in the affirmative. A written vote may be cast using standard mail, electronic mail, or facsimile. Upon approval of the motion, the entire Board or committee shall be notified by standard mail of the results of the vote.

Section 10. Committees of Directors. The Board of Directors may, by resolution adopted by a majority of the Board, designate one or more committees, including an executive committee, each committee to consist of two or more directors appointed by the Board. The Board may appoint one or more directors as alternate members of any committee, who may replace any absent or disqualified member at any meeting of the committee. Except as otherwise provided by the Rhode Island Non-profit Corporation Act or these Bylaws, any such committee, to the extent provided in the resolution, will have and may exercise all the authority of the Board of Directors; provided, however, that in the absence or disqualification of any member of such committee or committees, the member or members thereof present at any meeting and not disqualified from voting, whether or not he or she or they constitute a quorum, may unanimously appoint another director to act at the meeting in the place of any such absent or disqualified member. Such committee or committees will have such name or names as may be determined from time to time by resolution adopted by the Board of Directors. Each committee will keep regular minutes of its proceedings and report the same to the Board of Directors when required.

ARTICLE V
COMMITTEES

Section 1. Executive Committee. There will be established an Executive Committee which will consist of the President, plus three other directors. It will be the responsibility of the Executive Committee to adopt interim procedures and policies when necessary on behalf of TUG, subject to the ultimate approval of the Board of Directors.

Section 2. Technical Council. There shall be a Technical Council consisting of three directors, one of whom shall be appointed Technical Council Chair. The primary purpose of the Technical Council shall be the study of technical issues concerning TEX, Metafont, and their auxiliary support systems. The Council shall appoint Technical Working Groups to make recommendations related to specific issues. After evaluation and certification by the Council, the recommendations shall be brought to the Board of Directors for formal recognition. The Grand Wizard and the Wizard of Fonts shall be permanent honorary members of the Technical Council.

Section 3. Ad Hoc Committees. The Board of Directors may from time to time, by resolution adopted by a majority of the Board, appoint one or more Ad Hoc Committees to perform such functions as may be designated in said resolution.

Article VI
NOTICES

Section 1. How Delivered. Whenever under the provisions of the Rhode Island Non-profit Corporation Act or of the articles of incorporation or of these Bylaws written notice is required to be given to any person, such notice may be given by mail, addressed to such person at his or her address as it appears in the records of TUG, with postage thereon prepaid, and such notice will be deemed to be delivered, if mailed, at the time when the same will be deposited in the United States mail. Notice may also be given by telegram or personally to any director.

Section 2. Waivers of Notice. Whenever any notice is required to be given under the provisions of the Rhode Island Non-profit Corporation Act or the articles of incorporation or these Bylaws, a waiver thereof in writing, signed by the person or persons entitled to such notice, whether before or after the time stated therein, will be deemed equivalent to the giving of such notice. Attendance of a person at a meeting will constitute a waiver of notice of such meeting, except when the person attends a meeting for the express purpose of objecting to the transaction of any business because the meeting is not lawfully called or convened.

Section 3. Specification of Business. Neither the business to be transacted at, nor the purpose of, any meeting of the members of TUG or of a committee of the Board of Directors of TUG need be specified in any written waiver of notice except as otherwise herein expressly provided.

Article VII
OFFICERS

Section 1. Number. The officers of TUG will be a president, a vice president, a secretary, and a treasurer. The Board of Directors may from time to time elect or appoint such other officers including more vice presidents and assistant officers, as it may deem necessary. Any two or more offices may be held by the same person with the exception of the offices of president and secretary.

Section 2. Eligibility for Nomination. Any member may be nominated for TUG President, and any director may be nominated for the other Board offices.

Section 3. Nomination Procedure. Any member may be nominated for election to the office of TUG President by submitting a nomination petition in accordance with the TUG Election Procedures.

Section 4. Election and Term. The president will be elected by the general membership in accordance with the TUG Election Procedures.

All officers shall be selected for a term not to exceed two years. The terms of office and timings of elections may be lengthened or shortened by up to 9 months by the Board of Directors to avoid vacant posts or to avoid serious disruption to the business of TUG.

Officers other than the President shall be appointed by the Board of Directors. Any officer may be removed by the Board of Directors whenever, in its judgment, the best interests of TUG will be served thereby.

Section 5. Election Procedures. All elections will be conducted in accordance with Election Procedures approved by the Board of Directors. The election of the president shall be by written mail ballot of the entire membership. The candidate receiving the most votes will be elected.

Section 6. President. The President will preside at meetings of the General Membership, the Board of Directors and the Executive Committee.

Section 7. Vice President. The Vice President will serve in the absence of the President and will undertake other administrative duties as designated by the President.

Section 8. Secretary. The Secretary will maintain the records of TUG and see that all notices are duly given in accordance with the provisions of these

Bylaws or as required by law. The Secretary will also conduct TUG correspondence.

Section 9. Treasurer. The Treasurer will serve as chief financial officer and in general, will perform all of the duties incident to the office of Treasurer and such other duties as from time to time may be assigned to him by the President or Board of Directors.

Section 10. Vacancies. When an office becomes vacant for any reason, the President will appoint a director to serve out the remainder of that term. When the office of the President becomes vacant, the Vice President will become President for the remainder of the President's term and will then, as President, appoint a director to serve as Vice President.

Section 11. Signing of Instruments. All checks, drafts, orders, notes and other obligations of TUG for the payment of money, deeds, mortgages, leases, contracts, bonds and other corporate instruments may be signed by such officer or officers of TUG or by such other person or persons as may from time to time be designated by general or special vote of the Board of Directors.

Section 12. Voting of Securities. Except as the Board of Directors may generally or in particular cases otherwise specify, the president or the treasurer may on behalf of TUG vote or take any other action with respect to shares of stock or beneficial interest of any other corporation, or of any association, trust or firm, of which any securities are held by TUG, and may appoint any person or persons to act as proxy or attorney-in-fact for TUG, with or without power of substitution, at any meeting thereof.

Article VIII
EXECUTIVE DIRECTOR

Section 1. Duties. The Board of Directors shall select and employ an Executive Director who shall be responsible for the general administration of TUG's activities.

Section 2. Immediate Supervision. The Executive Director shall work under the immediate direction of the Executive Committee. The Executive Director shall attend meetings of the Executive Committee, the Finance Committee, and the Board of Directors, but shall not be a member of any of these bodies. The presiding officer of any of these meetings may request the absence of the Executive Director.

Article IX
SEAL

The corporate seal will have inscribed upon it the name of TUG and such other appropriate language as may be prescribed by the Rhode Island Nonprofit Corporation Act or from time to time by the Board of Directors.

Article X
FISCAL YEAR

The fiscal year of TUG will be determined by the Board of Directors and in the absence of such determination will be the calendar year.

Article XI
INDEMNIFICATION

Any person who at any time serves or has served as a director or officer of TUG or, at the request of TUG, as a director or officer of another corporation, whether for profit or not for profit, shall be indemnified by TUG and be reimbursed against and for expenses actually and necessarily incurred in connection with the defense or reasonable settlement of any action, suit, legal or administrative proceeding, whether civil, criminal, administrative or investigative, threatened, pending or completed, to which that person is made a party by reason of being or having been such director or officer, except in relation to matters as to which the person shall be adjudged in such action, suit or proceeding to be liable for negligence or misconduct in the performance of official duties. Such right of indemnification and reimbursement shall also extend to the personal representatives of any such person, and shall be in addition to and not in substitution for any other rights to which such person or personal representatives may now or hereafter be entitled by virtue of the provisions of applicable law or of any other agreement or vote of the Board of Directors, or otherwise.

Article XII
AMENDMENTS

The power to alter, amend or repeal the bylaws or to adopt new bylaws will be vested in the Board of Directors by affirmative vote of the directors in the manner provided in these Bylaws.

1996 Election Canceled

Board of Directors

Officers:

Michel Goossens (President, Jul 1995 – Jul 1997)
Judy Johnson (Vice-President)
Sebastian Rahtz (Secretary)
Mimi Jett (Treasurer)

Jan 1994 – Jul 1996:[1]
Mimi Burbank
Peter Flynn
Michael Ferguson
Tom Rokicki[2]
Norman Walsh[2]

Jul 1995 – Jul 1996:
Mimi Jett[4]

Jul 1996 – Jul 1999[5]
Mimi Burbank

Jan 1994 – Jul 1997:[1]
George Greenwade
Yannis Haralambous
Jon Radel
Sebastian Rahtz
[unfilled][3]

Jul 1995–1998:
Barbara Beeton
Karl Berry
Judy Johnson
Jiří Zlatuška

[1] After the 1993 fall election, board terms began in January 1994. Lots drawn to assign 5 members each to 1.5-, 2.5-, and 3.5-year terms, to achieve a 3-year term beginning/ending with the July board meetings (hence the references to both months in the list of terms of office).

[2] Appointed by the President, July 1995, to fill vacancies due to resignations by Jackie Damrau and Nico Poppelier

[3] Vacancy due to resignation by Luzia Dietsche, effective July 1995.

[4] Appointed by the President, July 1995, to fill a vacancy due to a shortfall in candidates for the Spring 1995 elections.

[5] Shortfall in candidates for the Spring 1996 election.

Tel.: +1 (415) 982-8449 `tug@tug.org` Fax: +1 (415) 982-8559

Source: /tex-archive/usergrps/tug/tugboard.tex on CTAN. Rev. March 96.

Institutional Members

The Aerospace Corporation,
El Segundo, California

American Mathematical Society,
Providence, Rhode Island

CNRS - IDRIS,
Orsay, France

CERN, *Geneva, Switzerland*

College of William & Mary,
Department of Computer Science,
Williamsburg, Virginia

Communications
Security Establishment,
Department of National Defence,
Ottawa, Ontario, Canada

CSTUG, *Praha, Czech Republic*

Elsevier Science Publishers B.V.,
Amsterdam, The Netherlands

Florida State University,
Supercomputer Computations
Research, *Tallahassee, Florida*

Grinnell College,
Noyce Computer Center,
Grinnell, Iowa

Hong Kong University of
Science and Technology,
Department of Computer Science,
Hong Kong

Institute for Advanced Study,
Princeton, New Jersey

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

Iowa State University,
Ames, Iowa

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

Marquette University,
Department of Mathematics,
Statistics and Computer Science,
Milwaukee, Wisconsin

Mathematical Reviews,
American Mathematical Society,
Ann Arbor, Michigan

New York University,
Academic Computing Facility,
New York, New York

Nippon Telegraph &
Telephone Corporation,
Basic Research Laboratories,
Tokyo, Japan

Princeton University,
Princeton, New Jersey

Smithsonian Astrophysical
Observatory, *Cambridge,
Massachusetts*

Space Telescope Science Institute,
Baltimore, Maryland

Springer-Verlag,
Heidelberg, Germany

Stanford University,
Computer Science Department,
Stanford, California

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

United States Naval Observatory,
Washington DC

University of California, Berkeley,
Center for EUV Astrophysics,
Berkeley, California

University of California, Irvine,
Information & Computer Science,
Irvine, California

University of Canterbury,
Christchurch, New Zealand

University College,
Cork, Ireland

University of Delaware,
Newark, Delaware

University of Groningen,
Groningen, The Netherlands

Universität Koblenz–Landau,
Koblenz, Germany

University of Manitoba,
Winnipeg, Manitoba

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

University of Stockholm,
Department of Mathematics,
Stockholm, Sweden

University of Texas at Austin,
Austin, Texas

Università degli Studi di Trieste,
Trieste, Italy

Uppsala University,
Uppsala, Sweden

Vrije Universiteit,
Amsterdam, The Netherlands

Wolters Kluwer,
Dordrecht, The Netherlands

Yale University,
Department of Computer Science,
New Haven, Connecticut

TEX Consulting & Production Services

Information about these services can be obtained from:

TEX Users Group
1850 Union Street, #1637
San Francisco, CA 94123, U.S.A.
Phone: +1 415 982-8449
Fax: +1 415 982-8559
Email: tug@tug.org

North America

Anagnostopoulos, Paul C.
Windfall Software,
433 Rutland Street, Carlisle, MA 01741;
(508) 371-2316; greek@windfall.com
We have been typesetting and composing high-quality books and technical Publications since 1989. Most of the books are produced with our own public-domain macro package, ZzTEX, but we consult on all aspects of TEX and book production. We can convert almost any electronic manuscript to TEX. We also develop book and electronic publishing software for DOS and Windows. I am a computer analyst with a Computer Science degree.

Cowan, Dr. Ray F.
141 Del Medio Ave. #134, Mountain View, CA 94040;
(415) 949-4911; rfc@netcom.com
Twelve Years of TEX and Related Software Consulting: Books, Documentation, Journals, and Newsletters
TEX & LATEX macropackages, graphics; PostScript language applications; device drivers; fonts; systems.

Hoenig, Alan
17 Bay Avenue, Huntington, NY 11743; (516) 385-0736
TEX typesetting services including complete book production; macro writing; individual and group TEX instruction.

NAR Associates
817 Holly Drive E. Rt. 10, Annapolis, MD 21401;
(410) 757-5724
Extensive long term experience in TEX book publishing with major publishers, working with authors or publishers to turn electronic copy into attractive books. We offer complete free lance production services, including design, copy editing, art sizing and layout, typesetting and repro production. We specialize in engineering, science, computers, computer graphics, aviation and medicine.

Ogawa, Arthur
40453 Cherokee Oaks Drive,
Three Rivers, CA 93271-9743;
(209) 561-4585
Experienced in book production, macro packages, programming, and consultation. Complete book production from computer-readable copy to camera-ready copy.

Quixote Digital Typography, Don Hosek
555 Guilford, Claremont, CA 91711;
(909) 621-1291; Fax: (909) 625-1342;
dhosek@quixote.com
Complete line of TEX, LATEX, and METAFONT services including custom LATEX style files, complete book production from manuscript to camera-ready copy; custom font and logo design; installation of customized TEX environments; phone consulting service; database applications and more. Call for a free estimate.

Richert, Norman
1614 Loch Lake Drive, El Lago, TX 77586;
(713) 326-2583
TEX macro consulting.

Type 2000
16 Madrona Avenue, Mill Valley, CA 94941;
(415) 388-8873; Fax: (415) 388-8865
pti@crl.com
$2.50 per page for 2000 DPI TEX and PostScript camera ready output! We provide high quality and fast turnaround to dozens of publishers, journals, authors and consultants who use TEX. Computer Modern, PostScript and METAFONT fonts available. We accept DVI and PostScript files only and output on RC paper. $2.25 per page for 100+ pages, $2.00 per page for 500+ pages; add $.50 per page for PostScript.

Outside North America

TypoTEX Ltd.
Electronical Publishing, Battyány u. 14. Budapest,
Hungary H-1015; (036) 11152 337
Editing and typesetting technical journals and books with TEX from manuscript to camera ready copy. Macro writing, font designing, TEX consulting and teaching.