

Baskerville

The Annals of the UK T_EX Users' Group

Editor: Editor: Sebastian Rahtz

Vol. 5 No. 1

ISSN 1354-5930

February 1998

Articles may be submitted via electronic mail to `baskerville@tex.ac.uk`, or on MSDOS-compatible discs, to Sebastian Rahtz, Elsevier Science Ltd, The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, to whom any correspondence concerning *Baskerville* should also be addressed.

This reprint of *Baskerville* is set in Times Roman, with Computer Modern Typewriter for literal text; the source is archived on CTAN in `usergrps/uktug`.

Back issues from the previous 12 months may be ordered from UKTUG for £2 each; earlier issues are archived on CTAN in `usergrps/uktug`.

Please send UKTUG subscriptions, and book or software orders, to Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Fax/telephone: 0121 476 2159. Email enquiries about UKTUG to `uktug-enquiries@tex.ac.uk`.

Contents

I	Editorial	3
1	Welcome back	3
2	<i>Baskerville</i> articles needed	3
3	A dozen from 1994: 9-3 to the good	3
4	News section	5
4.1	Winter L ^A T _E X Release	5
4.2	UKT _E X is dead, long live TeXhax	5
II	Letter to UKTUG	6
III	Omega: an extension of the T _E X System. March 16th, CERN	7
IV	The T _E X Users' Group goes to Florida!	8
V	Portable documents: Acrobat, SGML and T _E X	9
VI	MIDI2T _E X, a MusicT _E X tool	11
1	Introduction	11
2	MusicT _E X	11
3	What is wrong with it?	12
4	Other musical file formats, MIDI	13
5	Development of MIDI2T _E X	13
6	Portability	13
7	The program's features	14
8	Availability	14
9	The future	14
10	An example	14
VII	Maths in L ^A T _E X: Part 3, Different Sorts of Mathematical Object	17
1	Recall	17
2	Answers	17
2.1	Uneven subscripts	17
2.2	Roman text in notation	18
2.3	Spaces in subscripts	18
2.4	Empty set	18
5	A Spaced-out Interlude	18
5.1	Quads	19
5.2	Other Spaces	19
5.3	Phantoms	19

5.4	Horizontal Expanders	19
6	Exercises	20
7	Operators and relations	20
7.1	Limits	20
7.2	Operators	21
7.3	Novel uses of operators	21
7.4	Binary operators	22
7.5	Binary relations	22
7.6	Styles	22
8	Exercises	22
VIII	Elements of SGML	24
IX	Book Review — ‘Making T _E X Work’	26
X	Malcolm’s Gleanings	29
1	Whom the gods would destroy...	29
2	Garnered gleanings	29
3	Chimes at midnight	29
4	Message in a bottle	30
5	What next?	30
XI	Report of the 1994 UKTUG AGM	31
XII	Membership notes	34
1	Institutional Membership	34
2	UKTUG membership report	34
XIII	Chair’s report on 1993–94	35
XIV	The UK T _E X Users Group	36
1	Group aims and activities	36
2	Programme of meetings for 1995	36
XV	Obtaining T _E X	37

I Editorial

1 Welcome back

It is a pleasure to thank Robin Fairbairns for his sterling work in masterminding our special *Baskerville* 'FAQ' issue at the end of 1994. Many people assisted him in getting good, reliable, questions and answers, and we will maintain them in future. *Please*, therefore, report back to Robin any inaccuracies or omissions. We will reissue the list at the end of 1995, but in the meanwhile it is on the CTAN archives, on a WWW site, and in Acrobat format, for easy consultation. Contact a UKTUG committee member if you need help finding it.

2 *Baskerville* articles needed

As *Baskerville* starts a new year, determined to keep on coming out on time, your help is badly needed. It's time to write *Baskerville* articles for 1995. Get your writing hats on to 'delight fellow T_EX users with your words of wisdom'. *Please note the following copy deadlines:*

Issue	Submit rial	Last-minute notices	Posting date
5.2	Apr 10	Apr 14	May 5
5.3	Jun 5	Jun 9	Jun 23

Each issue of *Baskerville* has a special theme, although articles on any T_EX-related subject are always welcome. *Baskerville* 5.2 will have papers on SGML, Acrobat etc following the January meeting which is briefly reported in this issue.

3 A dozen from 1994: 9-3 to the good

Last year was a pretty good one for T_EX, I think. A lot of action, a lot of changes, and plenty of discussion. How can you keep up with it all? I find it a big job to scan the messages on various mailing lists, browse the Usenet `comp.text.tex` and read some T_EX publications; and I am not as assiduous as many. I do have an advantage over most people, though — I see all the Comprehensive T_EX Archive Network activity, as one of the small team of volunteer managers; just skimming the nightly logs of the mirroring process gives a fascinating insight into the busy world of developers continuously making new versions of things available; it's a rare day when there is *no* change in the archives. So maybe it will not be too unreasonable of me to attempt a review of the my favourites of 1994, and some of the turkeys. I'll take the chance to offload some favourite prejudiced opinions.

(+) **L^AT_EX 2_ε, of course** Aided and abetted by the 'three men and a dog' *L^AT_EX Companion*, L^AT_EX 2_ε has been a great success so far as I can tell. It was so undeniably needed, and so (largely) well done, that it has made L^AT_EX use a pleasure again. L^AT_EX 2_ε itself perhaps is most valuable for style designers and maintainers, but taken in conjunction with the *Companion*, there can be few users who do not find life a little easier this year. As an example of the hidden treasures, do you realize that if you use T1 encoding, and mark up accented words in the conventional way, they are translated on input into 8-bit codes and thus hyphenated?

(+) **NTG sensible article class** Who amongst you has never cursed the default L^AT_EX 'article' style? Have a look at the Dutch 'artikel' styles in the 'ntgclass' directory on the archives, recently updated for L^AT_EX 2_ε. I use them for lots of my work now, and find them very refreshing.

(+) **Eddi4T_EX, the corner shop T_EX shell** Ulrich Jahnze's shareware DOS shell for T_EX, Eddi4T_EX, is based around a friendly, but powerful, editor (the right way of going about it, in my view). It has extensible menus, a good macro language, online help, and is easy to maintain. Many have found that the superstore 4T_EX CD suits them, but I prefer something simple that doesn't run my life. This shell is admirably humble, but really does have the features you need.

(+) **MetaPost at last** Groupies of Donald Knuth will know that he swears by John Hobby's MetaPost¹ as his drawing tool of choice; this is a version of METAFONT which produces PostScript code rather than GF font format. It has been around for a few years, but just at the end of the year, AT&T finally allowed John to make it publicly available. Now everyone who is so inclined can use the enormously powerful METAFONT engine to describe their pictures (and use T_EX for labelling, of course) and get good portable output. John Hobby has recently written a graphing add-on, which is a joyful prospect for those of us who have been keening for *grap* ever since they stopped using *troff*.

(+) **Poor man's PostScript previewing** P. Pianowski and B. Jackowski have written a little set of PostScript procedures which magically add zooming, page selection, and so on to plain GhostScript; if you don't want to run up Windows, and its cranky Ghostview, then this is extraordinarily liberating. If you have never needed such things, forget you read this, but for cognoscenti, it's on CTAN in `support/psview`.

(+) **BaKoMa, or The Man from Protvino** I don't know how he does it, but the quality of Basil Malyshev's PostScript Type1 versions of the CM fonts, and the newly added AMS fonts (the BaKoMa family) is excellent, and means that I really can forget complicated METAFONT sources and weird PK fonts, without spending any money.

(+) **CD relief** I do a lot of T_EX at home, un-networked. How wonderful it has been to have the Prime Time Freeware CD of the CTAN archives to hand. No more directories full of 'just in case' copies. No more expensive and slow downloads of vital `.bst` files. OK, so some bits are out of date now, but there'll be a new release in the summer, so I can continue my more relaxed T_EX lifestyle.

(+) **HyperT_EX, and rich Acrobat from L^AT_EX** Adobe released version 2 of Acrobat this year, and their Portable Document Format; it coincided with maturing work by the loose HyperT_EX group (described by Arthur Smith in *Baskerville* 4.5), and the release of an enhanced *dvips* which can pass on hypertext information via the 'pdfmark' operator to PDF. This means that L^AT_EX can be set up to produce, automatically, very rich Acrobat documents. The whole HyperT_EX thing is encouraging, because it shows that there is plenty of life yet in T_EX.

(+) **The AMS** The American Mathematical Society has stuck by T_EX; they have kept up alongside L^AT_EX 2_ε with a major update to *AMS-L^AT_EX*; they document and support their macros and fonts. Better yet, they think creatively about the electronic future of scholarly publishing and maths, and look at SGML working with T_EX as the way forward. I hope they keep showing us the way.

(-) **T_EX under DOS** I am writing this article in a hotel in New York. Luckily, I was able to borrow a portable PC from work just before I left; but, my goodness, what a palaver it was to put together a useful T_EX system to take with me to prepare *Baskerville*! I ended up with:

1. An emT_EX kernel set of programs, and a L^AT_EX format;
2. the RSX loader to allow emT_EX to work in a Windows DOS box; usage requires only 10 years of experience;
3. *dvips*, which I had to recompile myself to get dynamic decompression of PostScript files;
4. GhostScript, to preview pages properly (I like to see my figures, my colours, my rotations etc — is that unreasonable?);
5. A set of Type1 PostScript fonts—Baskerville, Computer Modern, Monotype Times, Lucida (maths fonts for work), and (unreleased beta) DC fonts (we do *Baskerville* with T1 encoding, and use Computer Modern as the typewriter font);
6. Gnuemacs, to edit files in a civilised way; and Eddi4T_EX, as a DOS T_EX shell for when I felt strange;
7. Battalions of `sty`, `cls`, `fd`, `tfm`, and `vf` files I might end up pulling in by accident or design.

¹Oh, a pox on logos! No, this isn't set properly, because I never seem to have Knuth's slightly amended *logo* font to hand.

If only Y&Y drivers understood virtual fonts, or I had three weeks to spare sorting out the reencoding of my fonts to the bizarre Windows encoding, I'd have brought the excellent *diviwindo* as a previewer. But I still couldn't have previewed my pictures.

What a mess. It took me hours to make it all go together. Why can't I use, or recommend, an unequivocally good and complete T_EX for DOS and Windows, at any price?

(-) **T_EX user group politics** Like rats packed in a cage, T_EXies just can't stop mauling each other. TUG is rendered impotent by the inability of its directors² to agree on what to do about T_EX worldwide; DANTE withdrew its Special Director from TUG in pique at something no-one else understood; the French quarrel publicly about disk sets; the British slyly insert prima donna activists into all T_EX projects. Thousands of T_EX users and developers worldwide happily help each other in a great spirit of cooperation — why can't the user groups work together?

(-) **L^AT_EX 2_ε, I'm afraid** 'They' keep changing it, you know. Documents come out differently before and after Christmas, styles mysteriously crack in new places with brace fatigue; *AMS-L^AT_EX* appears and vanishes like the Cheshire cat. Colour madness permeates the kernel. Can the redoubtable Frank Mittelbach hold together his talented team?

4 News section

4.1 Winter L^AT_EX Release

The second full release of L^AT_EX 2_ε was placed on the CTAN archives on December 17th. This release fixes problems reported since June and incorporates the new 'inputenc' package as a standard method of supporting 8-bit input.

The document `ltnews02.tex` contains more details of the new features. See also the WWW page on <http://www.tex.ac.uk/CTAN/latex/>

L^AT_EX is available from CTAN hosts in the directory `macros/latex/base`, and a ready 'unpacked' version is available in `macros/latex/unpacked`. Note that this includes all the files in 'base' so you do not need both directories.

Unfortunately a couple of small bugs are present in the release. These are fixed in the file `macros/latex/base/ltpatch.ltx` which is automatically applied when you install.

4.2 UKTeX is dead, long live TeXhax

(David Osborne posted the following on 23rd December:)

With the welcome emergence of the Comprehensive T_EX Archive Network, there's no longer any need for a UK-oriented digest, originally created to announce developments of the UK T_EX Archive which Peter Abbott set up at Aston University. The digest was Peter's idea, modelled on the TeXhax Digest, and he acted as editor/moderator for the first few years of its existence. Looking back, I see I (DO) took over the editing of the digest with the V90 #21 issue and that from the beginning of 1987 to date, 336 digests were produced.

However, don't despair! The intention is to merge UKTeX into TeXhax, which will continue to act as a question, answer and announcement forum for the global T_EX community. The electronic distribution lists of the two digests will be merged and, for a while at least, contributions addressed to UKTeX will still be accepted but will appear in TeXhax, starting early in 1995. Initially, it's intended to produce TeXhax at weekly intervals, to maintain the timeliness of UKTeX for announcements, but this frequency may be modified in the light of experience, depending on the volume of contributions.

So, my sincere thanks go to all those who have encouraged and supported UKTeX in the past, particularly to Peter Abbott. I encourage your support of the TeXhax Digest and welcome your contributions to it in 1995 and beyond.

²The author of this piece is a director of TUG, and as guilty as others.

II Letter to UKTUG

The UKTUG committee thought it might be useful to publish the following letter from a member, and our response — Editor.

Dear Peter,

Thank you for the T_EX info and *Baskerville*.

I regret to say I shall not be renewing my membership of the UK Users' Group. While OzT_EX 1.7 has got round the problem of processing larger text files, it is so much slower (about four times) than OzT_EX 1.4 that I generally still use the old version. What a waste of disk space! Although T_EX is a wonderful piece of software and I use it every week, I am afraid that membership of UKTUG has been rather disappointing as it has not provided a single piece of information which is any use, such as why L^AT_EX 2_ε is so much slower, or what anyone is doing about it. Also, why is digit 8 in scriptsize (8) missing the top left hand corner? Why do we need all those PK-files with umpteen megabytes for each font size instead of using Mac outline font file? Why doesn't 'File/Send PostScript...' display replies from the printer (PostScript writestring, etc.)? How to display rotated text for individual items (not whole pages as OzT_EX User Guide, page 32)?

John Smeathers

The following reply was put together by Peter Abbott from input by the UKTUG committee.

We would suggest the last issue of *Baskerville* has provided *every* member (and I include myself) with valuable help and information.

L^AT_EX 2_ε has a large number of new features, and a much more flexible font selection scheme. Unfortunately, the price paid for the extra functionality is that L^AT_EX 2_ε uses more memory and is slower than L^AT_EX 2.09.

For documents in 2.09 compatibility mode, L^AT_EX 2_ε can be 75% slower than 2.09. For documents in 2_ε native mode, L^AT_EX 2_ε can be 25% slower than 2.09. These figures vary widely between documents, and L^AT_EX 2_ε performs much better on long documents than on short ones. So in summary, in the last 5 years, L^AT_EX has slowed by about 25% and uses about 15% more memory. This figures are much better than almost any commercially available package, and are caused by the new functionality L^AT_EX provides.

The L^AT_EX project team is aware that L^AT_EX has to run on machines with limited memory or processor power, and where possible they are making L^AT_EX more efficient. But during 1993–95, the L^AT_EX project team is responding to user demands for flexibility and functionality, and there will always be a compromise between responding to user demands for efficiency and for functionality.

OzT_EX can use any outline fonts, either commercial or public domain ones. You can get all the Computer Modern and AMS fonts in Type1 format in public domain versions. For rotated text, you will be able to use dvips, which comes with OzT_EX v1.8, if you renew membership. Alternatively you can obtain OzT_EX 1.8 from other sources and pay Andrew Trevorror direct the shareware fee. One of the benefits of membership of UKTUG is that when new versions are released they are distributed to UKTUG members who have paid the additional £5 fee (the first year fee is £30).

File/Send PostScript is provided to avoid having to purchase a utility. Normal printing does report back from the printer. The UK T_EX User Group has paid for a shareware licence for its members to use OzT_EX. The group does not have any responsibility for maintenance/updates or improvement to the software. In common with all application software, new releases bring new features and improvements which usually require more facilities such as memory or disc space. Likewise newer versions can be slower and indeed there have been major changes to the Mac operating system which are not penalty free.

Finally, the group has organised a number of meetings and other related activities which are open to members at preferential rates. There is of course the added bonus of discussing problems with others at meetings and the group actively supports the UK T_EX Archive which can be accessed by non connected members through me. The committee are continually looking at ways to improve the services offered to members.

III Omega: an extension of the T_EX System. March 16th, CERN

Dr. Michel Goossens
CN Division, CERN
CH-1211 Geneva 23,
Switzerland

Phone: (+41 22) 767-3363, Email: goossens@cern.ch

The T_EX program was originally developed in the mid seventies by Donald Knuth to typeset mathematical texts in the English language. Since then T_EX has made inroads in broader and broader areas of scientific, literary and other scholarly activities in many countries around the world. In 1991, Knuth froze T_EX, in the interest of stability, among other reasons. However, he allows the T_EX code to be used as the basis for further developments, so long as the resulting system is distributed under a different name.

The Ω package is an extension of T_EX developed by John Plaice (Université Laval, Canada) and Yannis Haralambous (Lille, France). Its first release, currently under beta-test, aims primarily at improving T_EX's multilingual abilities.

On Thursday, March 16th, 1995, Ω, the first generally available extended version of T_EX, will be presented to the world at CERN (Geneva, Switzerland). In Ω all characters and pointers into data-structures are 16-bit wide, instead of 8-bit, thereby eliminating many of the trivial limitations of T_EX. Ω also allows multiple input and output character sets, and uses programmable filters to translate from one to the other. Internally, Ω uses the universal 16-bit Unicode standard character set. These improvements will not only make it a lot easier for T_EX users to cope with multiple or complex languages (e.g., Khmer, Arab, Indic, Chinese, Japanese) in one document, but will also form the basis for future developments in other areas, such as native color support and hypertext features.

The standard distribution of Ω also includes the Unicode T_EX encoding, a new standard font encoding that is a superset of Unicode for — at this moment — the Latin, Greek, Cyrillic and Arabic character sets. The first font using this encoding, Universal Computer Modern, is based on existing METAFONT code prepared by different national T_EX users' groups. The UCM font is built using the Virtual METAFONT mechanism, which ensures that users can use the alphabets that they need without having to manipulate huge files.

The presentations on the 16th of March will take place in the CN Amphitheatre (Bld. 31, 3rd Floor). They will start with a general introduction of the Ω system, explaining how it is or can be implemented on various computer platforms. Then, the presentation of examples of high quality typography will show how the use of the Unicode/ISO 10646 character code makes it easier to handle a complete set of Cyrillic, Latin, Greek, Arab, Hebrew, and other characters in one document. At the end of the day a round-table discussion will allow the participants to provide input to the Ω team about which features they would like to see improved or added to the system. All participants will obtain a free copy of the software to take home with them. More technical discussions between implementors can be organized for the next day.

This day is sponsored by GUTenberg, the T_EX local user group of the French speaking community. Other sponsors would be much appreciated since we would like to invite as many representatives from different LUGs and language communities as possible to let everyone benefit from the possibilities of this new system for typesetting complex languages.

Information about Ω can be found in the Proceedings of the 1994 TUG Conference, *TUGboat* 15(3). Contact TUG at tug@tug.org for ordering these (over 240 page) proceedings if you are not (yet) a member of TUG. A somewhat updated version of the same material appeared in the Proceedings of the 8th European T_EX Conference in Gdansk, which can be obtained by sending 15 DM (postage included) to

Włodek Bzyl, Instytut Matematyki, Uniwersytet Gdański, Wita Stwosza 57, PL 80-952, Poland.

To register for the Ω day, or to receive further details, contact the author.

IV The T_EX Users' Group goes to Florida!

The T_EX Users' Group is proud to announce that the sixteenth annual meeting will be held at the TradeWinds Hotel, in St.Petersburg Beach Florida, July 24-28, 1995. We would like to extend a warm invitation to T_EX users around the world – come join us at one of the largest and most beautiful resort beaches in Florida, as we explore where T_EX is to be found and how its users are going far beyond—or are diverging from—its initial mathematical context.

The theme of the meeting will be “Real World T_EX” and this year we plan to have demonstrations of pre- and post-processors, and the active participation of developers and vendors, in hopes that you, the user, may discover “hands-on” just what can be done with T_EX and other utilities!

Commercial users of T_EX are particularly encouraged to attend. The meeting will feature papers of interest to publishers and T_EX vendors, a panel discussion addressing commercial users' needs and wants, and a gallery for displaying samples of T_EX work.

There will be the usual courses associated with the meeting: Intensive Courses in T_EX, L^AT_EX2e, PostScript, Graphics, and perhaps other topics. The meeting itself will have excellent speakers, panel discussions, workshops, poster displays, BoFs and technical demonstrations. A preliminary schedule will be forthcoming in February of 1995, so be sure to look for updates in TTN, TUGboat and on the WWW, at <http://www.ucc.ie/info/tex/tug/tug95sched.html>, and on the CTAN archives in [tex-archive/usergrps/tug](http://www.ctan.org/tex-archive/usergrps/tug) Nearer the time of the conference, there will be an online registration form located on the WWW at <http://www.ucc.ie/info/tex/tug/tug95form.html>

The TUG95 committee will be working with individuals who wish to share accommodations, as a way of helping defray expenses. The Bursary Fund is also available to assist users who demonstrate need. All members are encouraged to consider contributing to the fund. To obtain more information about contributing to or applying for the Bursary Fund, please contact the TUG office by email to tug@tug.org or by post to the address to

T_EX Users Group,
P.O. Box 869,
Santa Barbara, CA 93102-0869 USA.
tug@tug.org

Deadlines

Submission of Abstracts	January 31, 1995
Preliminary Papers Due	March 31, 1995
Other Proposals [†]	April 30, 1995
Preprint Deadline	June 23, 1995
Meeting Date	July 24 – July 28, 1995
Camera Ready Deadline	August 25, 1995

[†]Workshops, panels, posters, demonstrations, etc

Do you have suggestions or requests for topics to be covered? Then, please send email to tug95c@scri.fsu.edu, because this is your conference and we want to make it the “best ever”!

V Portable documents: Acrobat, SGML and T_EX

Lou Burnard

Oxford University Computing Service

lou@vax.ox.ac.uk

This joint meeting of the UK T_EX Users Group and the BCS Electronic Publishing Specialist Group in London on January 19th attracted a large and mixed audience of academics, T_EX hackers, publishers, and software developers, with representatives from most UK organizations active in the field of electronic publishing and document management. I was expecting rather more disagreement about the relative merits of the various approaches now available for the creation of portable documents; in the event, the path of SGML-based righteousness, with appropriate concessions to the practical merits of PostScript-based systems, was apparently endorsed by the consensus.

First of the seven speakers was David Brailsford from Nottingham University, who described Adobe's Acrobat as 'a *de facto* industry standard'. His presentation of exactly how the various components of this product worked together, and could be made to interact with both L^AT_EX and SGML, was very clear and refreshingly free of hype. The choice of PDF (which is effectively a searchable and structured form of Postscript, in which logical structure and hypertextual links are preserved along with the imaging information) as an archival format was a pragmatic one for journals such as EPodd where fidelity to every detail of presentation was crucial. The availability of a free Acrobat reader was also a plus point. He characterized the difficulties of mapping the logical links of a L^AT_EX or SGML document on to the physical links instantiated in a PDF document as a classic case of the importance of 'late binding', and revealed the open secret that Adobe's free PDF reader would soon be upgraded to recognise and act on HTML-style anchors. A demonstration of the Acrobat-based electronic journal project CAJUN is already available online at <http://quill.cs.nott.ac.uk/>

David Barron, from Southampton, gave an excellent overview of what exactly is implied by the phrase 'portable document'. Documents are not files, but compound objects, combining text, images and time-based media. There is a growing awareness that electronic resources should be regarded as virtual documents, repositories of information from which many different actual documents may be generated. These developments all make 'portability' (defined as the ability to render documents – with varying degrees of visual fidelity – in different hardware or software environments) very difficult. Portability was of crucial importance, not only for publishers wishing to distribute in the electronic medium, and not only for specific user communities wishing to pool information, but also for all of us. Information available only in a non-portable electronic form was information at the mercy of technological change. He cited as portability success stories the widespread use of PostScript and L^AT_EX as a distribution medium by the research community, referring to the Physics preprint library at Los Alamos as a case where this had now become the normal method of publication. By contrast, the success of the World Wide Web seemed to be partly due to its use of a single markup language (HTML) which effectively takes rendering concerns entirely out of the hands of authors. From the archival point of view, however, none of the available standards seemed a natural winner: hypertext was still too immature a technology, and there were still many intractable problems in handling multiple fonts and character sets. Professor Barron concluded with a brief summary of the merits of SGML as providing a formal, verifiable and portable definition for a document's structure, mentioning in passing that Southampton are developing a TEI-based document archive with conversion tools going in both directions between SGML and RTF, and SGML and L^AT_EX. Looking to the future, he saw the IBM/Apple Opendoc architecture as offering the promise of genuinely portable dynamic documents, which could be archived in an SGML form once static.

The third speaker of the morning, Jonathan Fine, began by insisting that the spaces between words were almost as important as the words themselves. I felt that he wasted rather a lot of his time on this point, as he did later on explaining how to pronounce 'T_EX' (surely unnecessary for this audience) before finally describing a product he is developing called 'Simsim' (Arabic for sesame, which is a trademark of British Petroleum we learned). This appears to be a set of T_EX macros for formatting SGML documents directly, using components of the ESIS to drive the formatter, but I did not come away with any clear sense of how his approach differed from that already fairly widely used elsewhere.

Peter Flynn, from University College Cork, did his usual excellent job of introducing the Wondrous Web World, focussing inevitably on some of its shortcomings from the wider SGML perspective, while holding out the promise that there is a real awareness of the need to address them. What the Web does best, in addition to storage and display of

portable documents, is to provide ways of hypertextually linking them. Its success raises important and difficult issues about the nature of publishing in the electronic age: who should control the content and appearance of documents – the user, the browser vendor, or the originator? Publishing on the Web also raises a whole range of fundamental and so far unresolved problems in the area of intellectual property rights, despite the availability of effective authentication and charging mechanisms. He highlighted some well-known ‘attitude’ problems – not only are most existing HTML documents invalid, but no-one really cares – and concluded that the availability of better browsers, capable of handling more sophisticated DTDs, needed to be combined with better training of the Web community for these to be resolved.

The three remaining presentations, we were told after a lunch in spartan surroundings, would focus on the real world, which seemed a little harsh on the previous speakers. Geeti Granger from John Wiley described the effect on a hard-pressed production department of going over to the use of SGML in the creation of an eight volume Chemical Encyclopaedia. Her main conclusions appeared to be that it had necessitated more managerial involvement than anticipated, largely because of the increased complexity of the production process. She attributed this partly to the need for document analysis, proper data flow procedures, progress reports etc., though why these should be a consequence of using SGML I did not fully understand. More persuasively, she reported the difficulty the project had had in finding SGML-aware suppliers, in designing a DTD in advance of the material it described, in agreeing on an appropriate level of encoding and in getting good quality typeset output.

Martin Key, from Elsevier, described in some detail the rationale and operation of the Computer Aided Production system used for Elsevier’s extensive stable of academic journals. Authors are encouraged to submit material in a variety of electronic forms, including L^AT_EX, for which Elsevier provide a generic style sheet. Other formats are converted and edited using an inhouse SGML-aware system (apparently implemented in WordPerfect 5, though I may have misheard this). This uses their own DTD, based on Majour, with extensions for maths, which seemed to be a major source of difficulty. Documents will be archived in SGML or PDF in something called an electronic warehouse, of which no details were vouchsafed. Both PDF and SGML were seen as entirely appropriate formats for online journals, CD-ROM and other forms of electronic delivery. The advantages of SGML lay in its independence of the vagaries of technological development, and its greater potential. However, potential benefits always had to be weighed against current costs; like any other business, Elsevier was not interested in experimentation for its own sake.

The last speaker was Michael Popham, formerly of the SGML Project at Exeter, and now of the CTI Centre for Textual Studies at Oxford. His presentation did a fairly thorough demolition job on the popular notion that there is still not much SGML-aware software in the world, starting with a useful overview of the SGML context – the ways in which SGML tools might fit into particular parts of an enterprise – and then listing a number of key products organized by category. It was nice to hear the names of so many real SGML products (auto-taggers, authoring aids, page layout systems, transformation tools, document management systems, browsers and parsers) being aired, after a long day obsessed by Acrobat and L^AT_EX. He concluded with a useful list of places where up-to-date product information can be found, and a reminder that the field is rapidly expanding, with new tools appearing all the time.

The day concluded with an informal panel session, onto which I was press ganged, which effectively prevented me from taking notes, but also gave me the chance to promote the recently-published DynaText version of the TEI Guidelines, which I did shamelessly. I also remember Malcolm Clark asking, tongue firmly in cheek, why everyone couldn’t just use Word, and being somewhat agreeably surprised by the number of people in the audience who were able to tell him the answer, and in no uncertain terms. Other topics addressed included auto-tagging, whether maths and formulae should be encoded descriptively or presentationally, whether Microsoft will still be around in the next century, and whether we would ever learn how to format documents for electronic presentation as well as we could on paper.

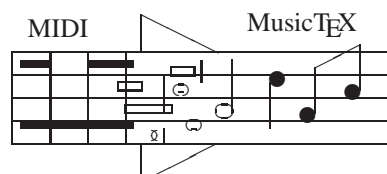
Lou Burnard
Oxford University Computing Service
lou@vax.ox.ac.uk

Editor’s note: Jonathan Fine contacted *Baskerville* to say that he feels Lou Burnard’s report (which was posted to the Text Encoding Initiative list) misrepresents his talk, misstating its central thesis regarding the space between words. Jonathan asks readers to reserve judgement until an article based on his talk is published in *Baskerville*. In the meantime he will send a copy of the OHP slides to anyone who asks.

VI MIDI₂T_EX, a Music_TE_X tool

Hans J.P. Kuykens
kuykens@amc.uva.nl

This article was first published in Dutch in the NTG journal MAPS in 1994. We thank the author and MAPS for allowing us to print it here.



1 Introduction

I must have been eight or nine years old. The NINT museum of technology in Amsterdam was my favourite because visitors were allowed to touch everything. The steering console of a trolley car was exciting because it made exactly the same rumbling sound when you used the throttle wheel. There were wheels to turn and knobs to push. My grandfather and I spent many hours strolling and playing in this museum. During one of our visits we did a guided tour through the computer department. In a separate room stood a giant, growling and buzzing monster. This beast still lived on punch cards. One of the more amusing demonstration programs controlled a line printer. By varying the typing speed it was possible to generate a tune. Prehistoric computer music, of course. However, this made such an impression on me... A computer that could sing, wow!

This example of computer music is of course nothing compared to the abilities that modern computers presently offer to the musician. Nowadays, the computer is applied in different ways in music. One application is *sequencing*: simply recording and playing tunes. The computer functions as a sort of merry-go-round organ. The organ-book is replaced by a computer file that stores the musical information. In this application the computer normally does not produce the sounds.

One step further the computer *is* responsible for the generation of the sound waves. In this case one speaks of *sampling*. Sound is digitized and recorded or played by the computer. If digitized sounds are mixed with the objective to create new sounds one speaks of *synthesis*. In the academic field computers are also used for *analysis* of the structure of music, mostly classical. How does Bach do it? Are there tricks or can we recognize typical patterns? Can we learn those tricks and teach a computer to automatically generate a similar composition? Thus, current computer technology can help the musician significantly with composition or performance of his musical ideas.

The computer has also proved to be a valuable tool for the generation of graphical output. This led to a revolution in the tools and methods used in the graphical industry. Nowadays, generation of sheet music can also be performed with a computer. Clear-cut programs are available, sometimes included with a sequencer program, for the development of sheet music. T_EX has proved to be a perfect tool for typesetting of text. However, this software package has more to offer and it appeared to be suitable for the layout of sheet music as well.

2 Music_TE_X

As T_EX became popular, it seemed that it might be suitable for ordering other things than text. Andrea Steinbach and Angelika Schofer [[3]] were the first who attempted to combine T_EX and music. They generated the necessary font files and a first set of macro files for the creation of sheet music using T_EX: MT_EX. The drawback of their system was that only a limited number of parallel staves could be generated. This must have been the motivation for Daniel Taupin, a physicist from France, to enhance this system. He based his work on the code of Steinbach and Schofer and developed a set of macro files that broke through the barriers of MT_EX. He named this package Music_TE_X [[4]]. Also Steinbach and Schofer continued to work on MT_EX and also this system has found its way to many users. However,

reprinted from Baskerville

Volume 5, Number 1

the enhanced flexibility of Music \TeX has led to a wider acceptance of this system. In recent years it has evolved into an extensive music notation package that allows not only to code traditional sheet music but also more exotic forms.

3 What is wrong with it?

When I first \TeX ed a Music \TeX file I was thrilled by the excellent quality of the output. Undoubtedly suitable for publications. Music \TeX , being built on \TeX , uses ASCII input files. These files store the codes that represent the musical information and here, to my opinion, lies the bottleneck of the system.

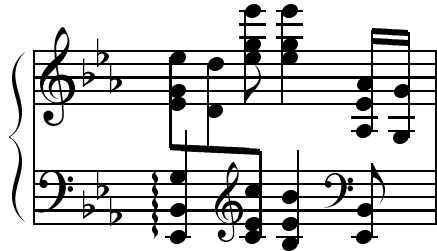
The computer keyboard is very suitable for the generation of a source text file for \TeX . If one types a ‘K’ in the input this character will also appear in the output. With sheet music this is quite a different story as there are no musical notes on the computer keyboard. In Music \TeX this problem is solved by coding the notes with \TeX commands. A note is characterized by two parameters: pitch and length. In sheet music pitch is represented by the vertical position of a note on a staff. The length of the note is specified by the type of symbol. For example, a semibreve is represented by an open oval without a stick. Coding a single note in Music \TeX thus requires two identifiers. This is solved by using a command that represents the length of the note (the type of symbol) and an additional parameter to specify pitch (the vertical position of the symbol on the staff). As an example the scale of C is coded. The first note is a semibreve and each subsequent note is half as long as the previous one.

```
\wh c \hu d \qu e \cu f \ccu g \cccu h \cccu i
```

this produces the following score :



Of course to code a more serious music piece we need more tools: i.e., multiple staves, different keys, incidentals, ornaments. To show that Music \TeX is also suitable to code more complex musical pieces an example of the Music \TeX manual is copied here (Brahm’s Intermezzo Op. 117,1, coded by Miguel Filgueiras).



This was coded as:

```
\begin{music}
\def\nbinstruments{1}\relax
\def\freqbarno{9999}% no bar numbers
\nbporteesi=2\relax
\cleftoksi={6000}\generalsignature{-3}% 3 flats
\debutextrait
\normal
\off{1em}\temps\Notes\larpeggio{E}5\zq{EI}\qu{N}%
\charnote0{\smallclefdesol}\relax
\nextstaff\qsk\ibl0e{-1}\zq{eg}\qb01\zq{d}\qb0k
\notes\cleftoksi={0000}\saveclefs\Notes\tbu0
\zq{ce}\qh0j\relax\nextstaff\zq{sn}\cl{1}\notes
\Notes\zq{be}\qu{i}\nextstaff\zq{sn}\ql{1}\notes
\cleftoksi={6000}\changeplefs\Notes\zq{E}\cu{I}%
\relax\nextstaff\ibbulh{-1}\zq{ae}\qhlh\bul
\zq{N}\qhlg\notes\finextrait
\end{music}
```

Coding a piece of music for Music \TeX thus involves editing an ASCII file with this type of commands. No-one

(musician or programmer) is going to take this in ‘at a glance’; the file is simply crammed with $\text{T}_{\text{E}}\text{X}$ commands; as a result, coding with Music $\text{T}_{\text{E}}\text{X}$ is a laborious and time consuming task. That’s what’s wrong with it.

4 Other musical file formats, MIDI

The computer has proved to be a powerful and versatile tool for sequencing. Many software packages are now available on most personal computers. Initially programmers of such software did not bother about standards of music files. You just had to stick to their product. Many different file formats occurred that were not exchangeable.

Along with the development of computers a growing interest occurred in electronic instruments. A tidal wave of keyboards and synthesizers was poured out over the market. The need occurred to connect instruments with each other and to computers to allow a full-size orchestra sound with only a few boxes of electronics. Different manufacturers’ instruments did not understand each other. Control of different companies’ sound modules was impossible or hardly possible.

This problem was soon recognized and about ten years ago musicians and manufacturers of electronic instruments formed a group to define some standards. MIDI was the result: Musical Instrument Digital Interface. Initially a communication protocol was developed that allowed electronic instruments to communicate with each other. MIDI was adopted by all companies and nowadays most keyboards and synthesizers are equipped with a MIDI interface.

Now that they were busy organizing standards this group also decided to tackle the problem of the many musical file formats. They developed a standard file format: the MIDI file. This file allows the exchange of musical pieces created with different programs.

There is no musical software company that is foolish enough to exclude support of this file format. Most sequencers now have the ability to import such MIDI files.

5 Development of MIDI2TEX

So, there I was. On the one hand a beautiful music notation package, Music $\text{T}_{\text{E}}\text{X}$, needing very complicated ASCII input. On the other hand MIDI files that could easily be recorded using a keyboard and sequencing program but lacking the ability to print nice sheet music. The link seemed so obvious. Surely, I thought, someone has already solved this problem with a program which translates MIDI files into Music $\text{T}_{\text{E}}\text{X}$ source code. I scanned the Internet but found nothing. There seemed to be no other alternative than to grab a chair, sit at my keyboard and code it myself. The first action was to get my hands on a good book about MIDI. I came home with: ‘MIDI programmer’s handbook’ by Steven De Furia and John Scacciaferro [[1]], an excellent book which explains the architecture in detail and contains many programming examples. I intended to develop just a small utility that would only translate the notes in the MIDI file into their Music $\text{T}_{\text{E}}\text{X}$ equivalents. I started programming on my Atari ST in Pascal. However, while developing the code the number of design specs grew. In the long term it evolved into a program that created a ready-to-compile Music $\text{T}_{\text{E}}\text{X}$ source file from a standard MIDI file. Command line parameters allowed altering specific options of the program.

I realized I had ended up with a piece of software that a lot of other computer music minded people would enjoy as well. A shame if it would die a lonely death in my computer. So, I decided to release the program as shareware in Cyberspace. In August 1992 I released version P 1.0. Initially, I distributed the package by email but when the number of requests exceeded one hundred (so people really *were* interested!) I decided to place the package on an FTP server. Of course many bug-reports followed that have led to further updates of the package.

6 Portability

The first versions of the program were only available for Atari ST and MS-DOS machines. Many requests reached me to port the code to other machines as well. However, Pascal is very poorly supported on other computer systems. If a Pascal compiler was available at all, it appeared that my source code was not compatible with that compiler. The programming language C is much more suitable for porting code to other computers due to standardization (ANSI-C). Many people inquired about the availability of C source code. That did not exist. At that moment the source had already grown to several thousand lines of code. My knowledge about C was minimal. So, translating the whole bunch into C by myself was not a realistic option. Finally, I learned that programs existed that could do a major part of the translation process automatically. Steve Chamberlain in the U.S. than poured all the Pascal source files into the P2C (Pascal to C) program. The resulting C source files than had been lying for a few months on a dusty shelf on the hard disk of my computer until I received a small note from Ad Verbruggen, who works at DIMES, Delft University.

He worked with a Macintosh, was familiar with C and most important he had the energy and spare time to teach the crippled C code to run. In concert with each other we worked out the C code until it functioned correctly on both the Mac and the PC. This version C 1.0 was released in the winter 1993–94. Immediately messages reached us that the program also compiled successfully on other systems. The goal was reached: MIDI2 \TeX was now available on most computer systems.

7 The program's features

Currently two versions are available: the C-version 1.3 of which only the source is available and P 1.3; a Pascal version. Before the release of version C 1.0 I could not resist to implement some new ideas in the Pascal version. In Pascal I program twice as fast as in C, for that reason the Pascal version was not yet put on a side track.

Version C 1.3 offers the following features :

Skip tracks Suppress output of specified tracks in the MIDI file.

combine tracks Two or more staves are combined as one instrument (for example left and right hand of a piano piece).

Bass clef The specified staves are coded using the bass clef.

A1.4 The specified staves are coded using the alto key (1..4).

Order Reorder the tracks in the MIDI file to the staves.

Quantize Quantization the start time and length of the notes in the MIDI file.

Key sign Specify the number of flats or sharps with which the piece is to be coded.

In addition, some layout parameters can be altered. Version P 1.3 offers the ability to automatically merge a text file into the generated code. MIDI2 \TeX is not a magic tool; it was developed simply to automate translation of the notes in a MIDI file into Music \TeX . In many cases it is still necessary to fine tune the resulting source code by hand. For a more extensive description of the program the reader should to read the manual that is included in the software package [[2]].

8 Availability

Music \TeX is freeware and available on the Internet through anonymous FTP from the CTAN archives. Both executables and C source code are available. The package will soon also be distributed on a number of shareware CD-ROMS including the 4 \TeX CD-ROM from NTG. Music \TeX has a discussion group on the Internet where information, problems and solutions are exchanged through email. Registration for this discussion group can be done by sending a request to `mutex-request.stolaf.edu`.

9 The future

The development of MIDI2 \TeX was performed on a fairly irregular basis. It is expected that this trend will continue. To further enhance portability to other systems some minor bugs will be removed. Due to the limited demands on the design the performance of the package is not optimal. Vertical alignment of notes is not always correct. Solving this problem is not so trivial and together with Ad Verbruggen and Dave Benson, a professor in mathematics in the USA, we are currently trying to find better algorithms.

The first versions of Music \TeX did not support the coding of triplets. As a result, this option was also not included in MIDI2 \TeX . In a future release the option to merge a text file will also be implemented in the C version.

There is a lot of work still to be done. With the ongoing development of Music \TeX new features become available: coding of percussion, guitar chords, etc.

10 An example

This article would not be complete without an example of the output of MIDI2 \TeX and Music \TeX , in Figure 1. In consideration of the available space only one page of Music \TeX source code is given below.

References

- [1] S. De Furia and J. Scacciaferro. *MIDI programmer's handbook*. M & T Publishing, Inc., 1989.

- [2] H.J.P. Kuykens. *MIDI2 \TeX , a MIDI to Music \TeX translation program*. Dapperstraat 273^c, 1093 BS Amsterdam, 1993. Email: kuykens@amc.uva.nl
- [3] A. Steinbach and A. Schofer. *M \TeX* . Master's thesis, Rheinische Friedrich-Wilhelms Universität, Bonn, Germany, 1987,1988.
- [4] Daniel Taupin. *Music \TeX , Using \TeX to write polyphonic or instrumental music*. Laboratoire de Physique ds Solides, Orsay, France, June 1992. (Version 4.52, (latest version is 5.07)).

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   MIDI2TeX C-version V 1.3 translation
%   of MIDI file :
%   VIVALDI.MID
%
%   Written by Hans Kuykens, Ad Verbruggen
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\normal
\hsize 160mm \vsize 240mm
\musicsize=16
\elemskip=6pt%
\def\nbinstruments{3}
\generalsignature{0}\relax
\def\thename{Vivaldi}
\medskip\centerline{\moyen \thename}
\rightline{translation by MIDI2TeX}
\rightline{by H.J.P. Kuykens}
\generalmeter{\meterfrac{4}{4}}%
\cleftoksi{{6}{0}{0}{0}}
\debutmorceau
%measure 1
\notes\qu {_I}&\zqp d\qup f&\ibbl1h{0}\qb1{_i}\qblh\notes\relax
\notes&&\qbli\tbl1\qblj\notes\relax
\notes\ibu0H{0}\qh0H&&\ibl2k{0}\qb2k\notes\relax
\notes\tbu0\qh0G&\zq g\cu {_i}&\tbl2\qb2{^k}\notes\relax
\tamps\notes\qu F&\zq h\ql j&\ql m\notes\relax
\notes\ds &\ds &\qs \ibbl3m{0}\qb3m\notes\relax
\notes\c1 M&\zq h\c1 j&\qb3n\tbl3\qb3o\notes\relax
%measure 2
\barre
\notes\ql {_b}&\zq k\ql {_i}&\ibbl3m{0}\qb3{_p}\qb3o\notes\relax
\notes&&\qb3n\tbl3\qb3m\notes\relax
\notes\qu I&\zq f\ibu2i{-1}\qh2i&\ibbl4i{0}\qb4{^k}\qb4{=k}\notes\relax
\notes&\zq d\tbu2\qh2f&\qb4j\tbl4\qb4i\notes\relax
\notes\ibl0{^K}{0}\qb0M&\zh c\hu f&\ibu5h{0}\qh5h\notes\relax
\notes\tbl0\qb0{^K}&&\tbu5\qh5g\notes\relax
\notes\ibu1{=K}{0}\qh1{=K}&&\qu f\notes\relax
\notes\tbu1\qh1J&&\notes\relax
%measure 3
\alaligne

```

Vivaldi

translation by MIDI2TeX
by H.J.P. Kuykens

The image displays a musical score for Vivaldi, consisting of five systems of three staves each. The notation includes treble, alto, and bass clefs, with various musical symbols such as notes, rests, and accidentals. Measure numbers are placed above the staves: 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14. The score is presented in a clean, black-and-white format, typical of a printed musical score.

Figure 1. MIDI2TeX and MusicTeX output

VII Maths in L^AT_EX: Part 3, Different Sorts of Mathematical Object

R. A. Bailey

Queen Mary and Westfield College, University of London

1 Recall

This is the third in a sequence of tutorials on typesetting Mathematics in L^AT_EX. The first two appeared in issues 4.4 and 4.5 of *Baskerville*. The series includes some things which can be found in [[5]], but I am working in more things which, while straightforward and necessary for Mathematical work, are not in [[5]] or [[6]]. In case you missed the first two tutorials, two warnings are now repeated.

I expect you, the reader, to do some work. Every so often comes a group of exercises, which you are supposed to do. Use L^AT_EX to typeset everything in the exercise except sentences in italics, which are instructions. If you are not satisfied that you can do the exercise, then tell me. Either write to me at

School of Mathematical Sciences
Queen Mary and Westfield College
Mile End Road
London E1 4NS

with hard copy of your input and output, or email me at `r.a.bailey@qmw.ac.uk` with a copy of the smallest possible piece of L^AT_EX input file that contains your attempt at the answer. In either case I will include a solution in the following issue of *Baskerville*: you will remain anonymous if you wish.

A word on the controversial issue of fonts. Fonts in Mathematics are handled differently in L^AT_EX 2.09, in NFSS, and in the new standard L^AT_EX, L^AT_EX 2_ε. Rather than compare these systems every time that I mention fonts, I usually limit myself to L^AT_EX 2.09. When you upgrade to L^AT_EX 2_ε, all these commands will still work, so long as you use the standard styles `article`, `report` and `book`. In the ‘Answers’ section below I expand a little on the dangers of using the font-changing commands given in [[6, Section 3.1]].

Many of the more complicated Mathematical things in this tutorial are not documented in [[5]] or in [[6]]. The L^AT_EX team warns me that they feel no obligation to support commands that are not in [[6]], so there is a danger that some of these things may change. However, everything given here works, in both L^AT_EX 2.09 and in L^AT_EX 2_ε, as at January 1995.

Some of the tricks described in this tutorial are at the edge of what you can conveniently do without using the `amstex` package. That package is undergoing change at the moment: I hope that by the time I reach the end of this sequence of tutorials the `amstex` package will have stabilized enough for someone to write an article explaining how to use it, including giving better methods than I can give here.

2 Answers

I promised to answer all questions arising from this series of articles (as far as I can).

2.1 Uneven subscripts

In *Baskerville* 4.5 Malcolm Clark asks about uneven baselines in subscripts. He gives a method of ensuring that all subscripts have the same baseline. I think that many Mathematical writers will not require that; nonetheless, some of us are uncomfortable with the unevenness in a single term such as

$$4z_1z_2^3$$

The easy way around this is to put a dummy superscript on the z_1 , because it is the superscript on the z_2 that is pushing the 2 down: thus

$$4 z_1^{\{ } z_2^3 \quad 4z_1z_2^3.$$

reprinted from *Baskerville*

Volume 5, Number 1

2.2 Roman text in notation

He also muses on whether to use `\textrm` or `\mathrm` or `\rm` in subscripts, if you are using L^AT_EX 2_ε. My advice is never to use `\textrm` in Mathematical notation. In the first place, `\textrm` does *not* give you roman type, according to such expert references as [[1, 2, 3]], all of whom say that ‘roman’ type is upright, as opposed to italic. All that `\textrm` does is give you back serifs and proportional spacing, if you had turned them off. Perhaps he meant `\textup`. But, secondly, I don’t think that you should use *any* of the commands `\text...` in Mathematical notation, because their effect depends on the surrounding text font but notation should be independent of the surrounding text. For example, try the following and compare the output:

```
{\rm $x_{\textup{big}} + \textup{size}_3$}
{\bf $x_{\textup{big}} + \textup{size}_3$}
{\bf $x_{\textrm{big}} + \textrm{size}_3$}
```

Malcolm was concerned because he wanted to obey the instruction in [[4]] to always use commands like `\textit{...}` rather than switches like `\it`. The trouble with that instruction is that the new commands `\text...` all work in a relative way. In my experience of writing (a lot of) Mathematics I have *never* needed such a relative change. I always need to specify my fonts absolutely, so that, say, the font chosen for long names of variables to be analysed does not change as the surrounding text font changes. Of course, it is sensible to do this with a macro such as `\variablename`; but that macro needs to call something with a syntax similar to `\textsl{...}` but which makes an absolute font change. I tried to persuade the L^AT_EX team to include commands like `\basesl{...}`, `\basett{...}` for such absolute changes, but I failed. Since the team wants to reserve the right to remove switches like `\tt` at some future time, this means that most of us will have to write our own macros, with our own idiosyncratic names, something like the following:

```
\DeclareTextFontCommand{\basett}%
  {\normalfont\ttfamily}
```

2.3 Spaces in subscripts

Malcolm also asked how to get spaces into subscripts. If I need to put a verbal phrase in a subscript then I use `\rm` and put in the interword spaces by hand.

```
\sum_{p{\rm\ is\ prime}} \frac{1}{p}
```

$$\sum_{p \text{ is prime}} \frac{1}{p}$$

2.4 Empty set

Kathleen Lyle has queried the symbol I gave last time for the empty set, with the command `\emptyset`. She points out that [[4]] shows a different symbol given by this command, a symbol which looks like a circle with a diagonal line through it and which is much closer to a Mathematician’s idea of the empty set than is \emptyset . But [[4]] also gives the command `\varnothing`, available with the package `amssymb`, which produces the symbol \emptyset . It appears that Knuth made a mistake in using the name `\emptyset` for the glyph which most of us think of as a variant form of zero. To correct this mistake, the AMS has redefined the command `\emptyset` to produce the symbol more like the empty set and given us `\varnothing` for the sake of those authors who really do want a zero with a line through it. It is a pity that [[4]] does not say that its `\emptyset` is the AMS one rather than the Knuth one.

What to do when a software author makes a mistake like this is a controversial question. Personally, much as I prefer the AMS’s empty set, I deplore such redefinition of a command, because it destroys portability of documents. Suppose that I write a document without the `amssymb` package and use `\emptyset`. I may send this document to someone (perhaps the AMS itself) who always uses the `amssymb` package when compiling documents. Even though I have made no explicit calls to commands defined by the package, my empty sets will come out looking different. A topologist may be content with the change; a computer scientist may not. In either case the document is printed with different symbols in the two cases, and this really should not happen. I think that it would have been better if the AMS had used a different name, such as `\trueemptyset`, for their empty set: then authors with access to the `amssymb` package could choose whether or not to include

```
\renewcommand{\emptyset}{\trueemptyset}
```

at the start of their files.

5 A Spaced-out Interlude

5.1 Quads

Traditionally, there are certain lengths of space (depending on the type size) which are always used in certain places in Mathematical typesetting. The most useful are the *quad* space and the two-quad space. When I was a copy-editor I used to just put the marks for these two types of space in the appropriate places in the copy; I did not have to know how big they were. Neither do you. In displayed Maths, use `\qqquad` to obtain a two-quad space between a formula and a short verbal condition or justification.

`y \in Y \qqquad\mbox{by defintion of~Y}`

If there are two short formulas linked in a display by a short verbal phrase (perhaps only one word) use `\quad` to produce a quad space on either side of the phrase.

`A \subteq B \quad\mbox{and}\quad A \ne B`

5.2 Other Spaces

A sequence of much smaller horizontal spaces that you can insert yourself is, in increasing order of magnitude,

`\, \: \; _`

They are called *thin space*, *medium space*, *thick space* and *interword space* respectively; their size also depends on the current type size. The thin space is usually needed after the `!` in factorials and often needed after a square root.

`\sqrt{3} \, , a \sqrt{3}a`
`5! \, , 4! \quad 5!4!`

It is also used before each *dx* term in an integral. On the other hand, in multiple integrals the integral signs may be too far apart, in which case the *negative* thin space `\!` should be inserted between them.

For consistency, these adjustments should all be made via macros. For example,

`\newcommand{\sqrtssp}[1]{\sqrt{#1}\,}`

will make `\sqrtssp` into the command for a square root with a little extra space, and a macro for factorials can be made similarly. For the integral signs you can use

`\newcommand{\intt}{\int\!}`

or the rather different solution provided in `amstex`. A suitable macro for the *dx* is

`\newcommand{\diff}[1]{\, d #1}`

which has the added advantage that if you believe that only variables should be in Maths italic then the `{\, d #1}` can be changed to `{\, {\rm d} #1}`.

5.3 Phantoms

The useful command `\phantom` allows you to leave a space whose horizontal and vertical dimensions are those of its argument. For example, if you want to define the notation `[]` as the least-integer function without specifying a dummy variable, you can type `[]`.

All digits are the same width, so `` produces a phantom digit. It is very useful in tables of data when all other methods of alignment fail. Make yourself a macro for it.

There are also horizontal and vertical phantoms `\hphantom` and `\vphantom` respectively. Each of these measures only one dimension of its argument.

5.4 Horizontal Expanders

In the first tutorial we saw that `\widehat` and `\widetilde` expand as far as necessary (up to a given upper bound) to cover their arguments. The following commands also expand horizontally to match the arguments:

`\overline` `\underline`
`\overrightarrow` `\overleftarrow`
`\overbrace` `\underbrace`

You can use a superscript to put a label on an overbrace, and a subscript with an underbrace.

`n\bar{y}^2 +`
`\overbrace{(y_1-\bar{y})^2 + \cdots +`
`(y_n-\bar{y})^2}^{\rm sum\ of\ squares}`

$$n\bar{y}^2 + \overbrace{(y_1 - \bar{y})^2 + \cdots + (y_n - \bar{y})^2}$$

6 Exercises

Exercise 23

$$(x_1 + x_2)^3 = x_1^3 + 3x_1^2x_2 + 3x_1x_2^2 + x_2^3$$

Exercise 24

$$\sum_{n \text{ divides } 10} n = 18$$

Exercise 25 In geometry, $\vec{AB} + \vec{BC} = \vec{AC}$.

Exercise 26 We define P_g by

$$t(vP_g) = (t^{g^{-1}})v \quad \text{for } v \in \mathbf{R}^T.$$

Exercise 27

$$2^a \times 2^b = \underbrace{2 \times \cdots \times 2}_a \times \underbrace{2 \times \cdots \times 2}_b = 2^{a+b}.$$

Exercise 28 An inner product $\langle \cdot, \cdot \rangle$ is defined on G^* by

$$\langle \theta, \phi \rangle = \frac{1}{|G|} \sum_{g \in G} \theta(g) \bar{\phi}(g).$$

Exercise 29 If $\bar{\cdot}$ denotes complex conjugation, then

$$\overline{\xi + \zeta} = \bar{\xi} + \bar{\zeta} \quad \text{and} \quad \overline{\xi \zeta} = \bar{\xi} \bar{\zeta}.$$

Exercise 30

$$\iint \phi(r, \theta) dr d\theta$$

Exercise 31

$${}^6C_2 = \frac{6!}{4!2!}$$

7 Operators and relations

7.1 Limits

In the second tutorial I introduced various things that could have their limits, or ranges, typed in as sub- and superscripts: standard functions with English names, like `\log`; repeated binary operators, like `\sum`; and the integral sign `\int`. \TeX thinks of all of these as operators. Some operators have the limits set above and below in displayed Maths, but to the right in text; others always have their limits set to the right. You can override these defaults by using one of the commands `\limits`, `\nolimits`, `\displaylimits` after the name of the operator. The integral sign normally has its limits set to the right: if you want them set above and below type `\int\limits`.

$$\int_0^2 x^3 \, dx = 4 \qquad \int_0^2 x^3 dx = 4$$

$$\int\limits_0^2 x^3 \, dx = 4 \qquad \int_0^2 x^3 dx = 4$$

If you want the limits to be above and below if the operator happens to be in displayed Maths, but to the right otherwise, use `\displaylimits` instead of `\limits`. Finally, to ensure that the limits always come to the right, use `\nolimits`.

If you want to change the size of the operator as well as the position of its limits, you probably need to see the section on styles below.

7.2 Operators

The standard functions with English names already provided by T_EX cannot be enough for the whole of Mathematics. You make new ones by using `\mathop`, usually inside a `\newcommand`. For example,

```
\newcommand{\var}{\mathop{\rm Var}\nolimits}
\var X \geq 0
```

$$\text{Var}X \geq 0$$

(If you have L^AT_EX 2_ε, you may feel safer using `{\mathrm{Var}}` in place of `{\rm Var}`.) You may put one of `\limits`, `\nolimits`, `\displaylimits` after the contents of the `\mathop`, to specify how sub- and superscripts should behave. Putting nothing is equivalent to putting `\displaylimits`.

There is a school of thought that all operators should be in the same font, so that the `\rm` in the definition of `\var` should be replaced by a command like `\operatorfont`, which would, of course, be defined in the style file or in the preamble to the document. I do not agree with this. It is not at all unusual to use bold for the expectation operator while retaining roman for the variance.

If you make a single letter into an operator, it will be vertically centred, which may not be what you intend:

```
\newcommand{\ee}{\mathop{\rm E}\nolimits}
\[\ee X + \ee Y = \ee(X+Y)\]
```

$$EX + EY = E(X + Y)$$

To override this, put the single letter in a box:

```
... \mathop{\mbox{\rm E}} ...
```

7.3 Novel uses of operators

In the first tutorial I said that you did not need to think of the symbol `'` as a superscript. Usually you do not, but T_EX always does, so you occasionally get unexpected results. You might want to write \sum' for a variant of the usual summation, perhaps to indicate omission of all i for which $\lambda_i = 0$, as in

$$\sum'_{i=1}^n \frac{1}{\lambda_i} P_i.$$

If you use `\sum'` it will come out as

$$\sum'$$

in display, and even worse things happen when you try to put limits on. Writing `\sum\nolimits'` cures the problem about placing the dash, but then you no longer have an operator to put limits on. So you need to make the whole of \sum' into an operator:

```
\newcommand{\sumsum}{\mathop{\sum\nolimits'}}
```

```
\[\sumsum_{i=3}^7\]
```

$$\sum'_{i=3}^7$$

If you look closely you will now see that the limits are centred on the whole of \sum' . This is logical, but may not be exactly what you intended. I do not know how to do the illogical but more aesthetically pleasing version, but a method is provided in `amstex`.

Sometimes you want to put a range of summation under (or over) the middle of a pair of summation signs. Do this by turning the pair of summation signs into an operator:

```
\newcommand{\twosum}{\mathop{\sum\sum}}
```

```
\[\twosum_{1<i<j<n} x_i x_j\]
```

$$\sum\sum_{1<i<j<n} x_i x_j$$

To get two ranges of summation under a summation sign, make an operator containing the summation sign and the interior range(s), and then put a subscript on that:

`\[\mathop{\sum_{j=1}^n}_{j\neq i} Y_j\]`

$$\sum_{\substack{j=1 \\ j\neq i}}^n Y_j$$

You would normally do this only in displayed Maths.

7.4 Binary operators

\TeX does not class ordinary binary operators as operators. Use `\mathbin` to make something into an infix binary operator. For example, `\mathbin{**}r` gives $n**r$. (What does `\mathbin{**}r` produce?) Usually this is done with a `\newcommand`. Even a single symbol may need to be explicitly turned into a `\mathbin`, if it is not one already, so that the spacing and linebreaks around it are correct: this is as true for single symbols that already exist as for those that you build up laboriously out of pieces. If the new binary operator is (part of) an English word, you will need to specify the font, just as for `\mathop`.

7.5 Binary relations

In the same way, `\mathrel` is used to make new binary relations. The considerations are similar to those for `\mathbin`s. Note that `\mathbin`s and `\mathrel`s are different in several subtle ways, such as the spacing around them, the linebreaks near them, and their behaviour when they do not find themselves between two ordinary symbols (compare `\mathbin{**}r` with `\mathrel{**}r`). If you are not a Mathematician you will probably have to ask the author of the document whether a particular squiggle is an operator or a relation.

If the new relation consists of two parts, one on top of the other, you can make the new relation in one step with `\stackrel`.

$$\Phi \stackrel{\text{rev}}{\sim} \Psi$$

7.6 Styles

When `\sum` appears in displayed Maths it is larger than in text Maths, and has its limits in a different place. However, once it is inside a fraction or an array, even in displayed Maths, it reverts to its appearance in text Maths. To force one style or the other, precede `\sum` with either `\displaystyle` or `\textstyle`.

$$\left[\frac{\sum_i x_i}{n}\right]$$

$$\left[\frac{\displaystyle\sum_i x_i}{n}\right]$$

$$\frac{\sum_i x_i}{n}$$

These two commands can affect the appearance of many items in Maths mode, including `\frac`.

There are analogous commands `\scriptstyle`, which sets the following items as if they were in a subscript, and `\scriptscriptstyle`, which sets them as if they were in a second-level subscript.

None of these four commands takes an argument. They are all switches, like `\rm` and `\large`, and apply until the end of the current subformula (such as the numerator of a `\frac`).

8 Exercises

Exercise 32 If f is a probability density function then

$$\int_{-\infty}^{\infty} f = 1.$$

Exercise 33 We assume that Y is a random vector with

$$\text{Cov} Y = \sum_{\alpha \in A} \xi_{\alpha} S_{\alpha},$$

where the S_{α} are known symmetric matrices satisfying $S_{\alpha} S_{\beta} = \delta_{\alpha\beta} S_{\alpha}$ and $\sum_{\alpha \in A} S_{\alpha} = I$.

Exercise 34 The definition of variance is: $\text{Var} X = \mathbf{E}(X - \mathbf{E}X)^2$.

Exercise 35 The optimize function `opt` is defined so that $\text{opt}M_i$ is equal to $\max\{\max_{i=1}^n M_i, 0\}$.

Exercise 36

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n y_i y_j = \left(\sum_{i=1}^n y_i \right)^2 - \sum_{i=1}^n y_i^2.$$

Exercise 37

$$\sum_{\substack{0 < i < m \\ 0 < j < n}} P(i, j).$$

Exercise 38 Define the operator $\boxed{+}$ on finite sets of integers as follows. If A and B are two finite sets of integers, then $A \boxed{+} B$ is the multiset of integers in which the number of times that n occurs is equal to

$$|\{(a, b) : a \in A, b \in B, a + b = n\}|.$$

Exercise 39 We want to write our wreath products in reverse order, so we put $G \text{rw} H = H \wr G$.

Exercise 40 The relation ρ is said to be symmetric if

$$x \rho y \text{ implies } y \rho x.$$

Exercise 41 The strong law of large numbers states that if X_1, X_2, \dots are independent and identically distributed with finite fourth moment then

$$\frac{X_1 + \dots + X_n}{n} \xrightarrow{\text{a.s.}} Y,$$

where $\Pr[Y = E(X_1)] = 1$.

Exercise 42 Define the relation $<_2$ on the natural numbers by $n <_2 m$ if $n \mid m$ and m/n is odd. This is a partial order.

Exercise 43 Create a binary operator for the colon in $G:H$ and compare it with `:` and `\colon`.

Exercise 44 (Redo Exercise 16 with a built-up fraction instead of the solidus, with the large operators remaining the same size.)

$$\prod_{k \geq 0} \frac{1}{(1 - q^k z)} = \frac{\sum_{n \geq 0} z^n}{\prod_{1 \leq k \leq n} (1 - q^k)}$$

Exercise 45 Redo Question 41 with the ‘a.s.’ in normal-sized type.

Exercise 46 More on binomial coefficients:

$$\sum_{\substack{1 \leq n \leq m \\ 1 \leq r \leq n}} \frac{n!}{r!(n-r)!} = \sum_{n=1}^m 2^n = 2^{m+1} - 2.$$

Exercise 47

$$\prod_{i=0}^m f(\lambda_i)$$

References

- [1] *Hart's Rules for Compositors and Readers*, Oxford University Press, Oxford, (1967).
- [2] *The Chambers Dictionary*, Chambers Harrap, Edinburgh, (1993).
- [3] *The Chicago Manual of Style*, The University of Chicago Press, Chicago, (1982).
- [4] Goossens, M., Mittelbach, F. & Samarin, A.: *The L^AT_EX Companion*, Addison-Wesley, Reading, Mass., (1994).
- [5] Lamport, L.: *L^AT_EX: A Document Preparation System*, first edition, Addison-Wesley, Reading, Mass., (1986).
- [6] Lamport, L.: *L^AT_EX: A Document Preparation System*, second edition, Addison-Wesley, Reading, Mass., (1994).

VIII Elements of SGML

Jonathan Fine

I don't have time to write a proper column for this issue, but I did not want too long to go by before giving the answers to the exercises at the last of my last article, in Baskerville 4(5). So here are the answers.

When was the SGML standard published? The standard was published in 1986. Although an international standard, it is still far from being a *de facto* standard in most of areas of application. One reason for this is resistance from the owners of proprietary document formats, as used by word processors, desktop and larger publishing systems and so forth.

Would it be a sound commercial move for the owners of, say *Word* to open it up so that it could be used with someone else's formatting software? Such as \TeX equipped with suitable macros? Should such happen then the next thing you know someone else will be bring out a word processor front-end *without* any formatting capabilities at all, but relying on the back end previously provided. And before you know where you are, *Word* will be just one of a number of competing software products all capable of operating on the same documents. *Word* is a registered trademark of Microsoft. At Bridewell I asked the panel what must fail if SGML is to succeed, to which Lou Burnard replied "Proprietary data formats," which answer I cannot improve upon.

Who are the five persons involved in text processing? Who is the sixth? In the article I listed author, designer, typesetter, implementor and publisher. The sixth was the reader. Each has a different interest in the document. Probably this list is a simplification. Geeti Granger in her talk at Bridewell (see elsewhere in this issue for a brief summary) described the production process for an encyclopedia of Chemistry that John Wiley has recently produced. This occupied two detailed transparencies. There are also editor, subeditor, proofreaders, picture editors and so forth. It is traditional, and for good reason, when establishing a database to spend a period analysing the size and structure of the data to be stored, and the operations that will need to be performed on it. And this before writing any but prototype code.

It seems sensible to follow this tradition when building a publishing application based on SGML. This design should be performed by an expert (either hire one, or become one yourself, or at least follow someone else's good example) in the documents you are hoping to create, store and process using SGML and SGML-compliant software. Just as with other computer applications, beware of trusting the technical expert in the language who does not understand the activity whose facilitation is intended. This applies double to applications which involve significant interaction among people and thus to SGML applications almost without exception. Deciding on data-storage standards for text is considerably more subtle than the design of a business order processing database. And that in itself is easy enough to get wrong. Good experts in SGML document design (and I am not one) are worth good money.

How does SGML differ from other computer languages? So far as I know, the purpose of all other computer languages is to describe in one way or another what it is that a computer should do. In other words they are languages for the writing of programmes. SGML is a language for the creation (and storage) of structured documents. There is no sense in SGML of anything doing anything. To use the technical terms, it is all syntax and no semantics. (Some complain that with \TeX it is the other way around.)

SGML is deliberately neutral as to what might be done to the document. It may be stored, spell-checked, formatted, edited or deleted. Or transmitted across the World Wide Web. All this is none of the concern of SGML, although needs must be carefully considered and specified if an SGML-based application is to be successful. I like to think of SGML as being ASCII for structured documents. And is ASCII a computer language?

What is the purpose of the prolog? Who writes it? The prolog contains the document type declaration, which specifies the markup elements and in which combinations, that can occur in the document instance. Technically an SGML document consists of a prolog and a document instance conforming to the markup declared in the prolog. Very often

```
<!DOCTYPE message SYSTEM "message.dtd" >
```

is used to refer to an external entity which contains the element and entity declarations. All this is a little technical and should be hidden from authors (and readers).

What is the purpose of the SGML declaration? Usually the SGML declaration is implied, which means that the software will supply or assume that a standard SGML declaration is to be used or will be applied. However it does

contains important information about how the following document is to be parsed. For example, it indicates whether omitted tags, short references, or other markup minimization devices may be used. It also declares whether the case is significant in entity and tag names. It also gives capacity information, such as the maximum allowed length of identifiers. In short, it sets all manner of switches that control the parsing of the following document, and indicates to the parsing software what capabilities it will require. An SGML declaration can also be written for parsing software, which indicates the documents it is capable of parsing.

What is the purpose of the document instance? Who writes it? At last! The author’s words, written by the author (or in some cases a ghost writer).

How many lines in a sonnet? Is that every sonnet? Trick question. Most sonnets have fourteen lines, but Shakespeare’s sonnet 126 (O thou my lovely boy, who in thy power) has but twelve. The pedants (oops, experts) can argue as to whether it really is then a sonnet, but when all is done, it had better appear in the appropriate place when Shakespeare’s sonnets are stored or printed. An old adage has it that altogether most of a computer program is devoted to dealing with the exceptions. Or to put it another way, the devil hides in the details. This is one of the attractions of `tex` the program. There’s almost nowhere for the bugs to hide.

What is the difference between mark up and markup? An easy one. The first is a verb, the second a noun. The words *mark down* and *markdown* have a similar relation although quite different meanings.

What is a declaration? In SGML markup constructions are declared, which then makes them available for use. There is an SGML declaration (which makes SGML available one assumes), and then a document type declaration which in turn contains element, entity and attribute declarations. Other features of SGML are also activated via declarations. Even comments are enabled as text within comment declarations, according to the terminology of the standard.

How is SGML a compromise? Between whom? Is this good or bad? A complete answer to this question would be the whole past, present and future history of SGML. It is a document standard, so the more features and power it provides the happier those who store documents will be. But if too much power is required, the implementors will be scared off or, perhaps worse, implement only an idiosyncratic subset of the language. Perhaps part of the success of HTML is due to it being strong enough to be useful to users, and weak enough to be implemented by developers.

If too much is required of authors they will simply not use it. Thus, a considerable part of the language is concerned with making life easier for the author to create a valid SGML document instance using an ordinary text editor. One of the functions of an SGML parser is to relieve the application programmer of the burden of accomodating (and perhaps getting wrong) these author-friendly features. The question of the allowable content models (a topic we have not yet discussed) has been a subject of much complaint and disagreement from and amongst implementors and document type designers. This question is the subject of an Annex to the International Standard.

(Experts only) In explaining SGML for beginners, have I made any false statements? Almost certainly yes. Mostly omissions I hope, and only the whitest of lies. I would welcome hearing of any corrections that might help beginners.

IX Book Review — ‘Making TeX Work’

Malcolm Clark
m.clark@warwick.ac.uk

```
@book{Walsh94,  
  title="Making TeX Work",  
  author="Norman Walsh",  
  publisher="O'Reilly &  
    Associates, Inc.",  
  address="Sebastopol, CA",  
  ISBN="1-56592-051-1",  
  pages=483}
```

This is truly a niche market book. Until I read it, I didn't realise there was a need for it. Basically it tries to describe the many tools and tool fragments which exist for use with TeX (and METAFONT). In general terms we are talking of tools available through the Internet, and most probably those available at your nearest convenient CTAN host. It is therefore encyclopedic rather than especially profound. You are unlikely to gain any new insights into the working of TeX and its friends, but you will probably meet lots of new friends, whose existence you never even imagined (I excuse a couple of people from this: almost inevitably our esteemed editor will be aware of everything mentioned here, and probably has a list of other tools which should have been included). Given the nature of the Internet world, and TeX, this can be no more than a snapshot which is probably already out of date: for example, the IP address for `ftp.tex.ac.uk` has changed.³

The book was first published in April 1994, and by July had been reprinted with ‘minor corrections’. The reprinted edition was one of the goodies at last year's TUG conference in Santa Barbara. Thank you O'Reilly & Associates, Inc.

Norm describes CTAN as ‘a fully-mirrored anonymous FTP hierarchy on three continents’ – but the three continents are North America, Europe and, er, Europe. He makes the common mistake of equivalencing UK with England. World Wide Web makes an appearance indirectly, rather than as a CTAN supported service (so much easier to use); to be fair he does mention the WWW interface available through O'Reilly's server,

<http://jasper.ora.com/CTAN/ctan.html>

If you try this URL⁴, you will find it would have been better to go straight to

<http://jasper.ora.com/ctan.html>

You may be concluding by now that we are strongly oriented towards electronic networks. This is true, although Norm does list some ways of obtaining distributions through more traditional means. But imagine you wanted to get hold of emTeX or OzTeX: Norm merely refers you to CTAN for emTeX and omits any contact at all for OzTeX. This despite the trailer at the end of issues of TeXhax with details of how to obtain this software on disc. Similarly he seems to have missed the (admittedly fairly recent) CD-ROM offerings from both the Dutch-speaking users group and from PrimeTime Software which contain gobs of material useful to those with and without electronic connection – let's not forget that those of us on SuperJanet are still a minority. Many TeX users still access a network (if they access one at all) over a telephone line, and the thought of downloading all these files, even if you can locate them, is daunting.

We are also fairly Unix oriented. This is hardly surprising, given that this is an O'Reilly book, and they have made their name by producing books, which if not 100% Unix in flavour, are at least very oriented towards Unix and GNU/Free Software Foundation. Thus there is a sort of assumption that you can probably write *Perl*, and fairly readily modify the *Perl* scripts which Norm includes. He uses *Perl* because it is a scripting language available on all the platforms he discusses – Unix, MSDOS (and Windows), OS/2 and the Macintosh. Sadly he doesn't tell me how to get hold of *Perl*. It is this Unix bias which leads him to assert that ‘Unix is probably the most common TeX platform’

³For the curious, it is now 128.232.1.87 — Editor

⁴If you are confused by all this stuff, help may be at hand. We hope that a future edition of *Baskerville* will discuss the TeX resources available through World Wide Web, and give some clear pointers, as well as filling in some of the background.

and to go on to note that the emphasis of the book is Unix workstations running X11 producing output for PostScript and HP LaserJet printers.

Norm suggests that his description of T_EX is really for the systems administrator who wants to know a bit more about this behemoth which he has been persuaded to let occupy her or his system. The computer naive, or just those wanting to get on with some documents are given scant regard. Although he does include some discussion of MSDOS and of OS/2, this is really because many tools have been ported to these platforms from Unix. The inclusion of the Macintosh is intriguing, but other than ghettoising it by giving it its own exclusive chapter out of the sixteen, it makes little appearance, and the repeated themes of command lines and restricted file names does demonstrate that it was something of an afterthought. The really useful fact of Mac filenames – that they should not contain a blank space if you hope to use L^AT_EX, is omitted.

In general, there also seems to be a bias towards T_EX, rather than the more accessible extensions like L^AT_EX, although again Norm does mention L^AT_EX, Lollipop (a little surprisingly) and Texinfo (unsurprisingly, although again demonstrating the Unix bias).

I have some misgivings when I start with Norm’s description ‘What is T_EX’. The first aesthetic principle which he ascribes to T_EX is: ‘The right margin is justified’ (page 3). Besides being a rather contested aesthetic principle, T_EX of course allows you to set ragged right, or ragged left (or even both). He also claims ‘ladders are avoided’. None of my books on typography mention ladders: I assume them to be equivalent to rivers, and while this is certainly a claim by Knuth & Plass, and I have yet to see rivers created by T_EX, I’m still not convinced of its universal truth.

I’m made uncomfortable by the first table (page 9), where among the special characters for plain T_EX, Norm suggests using < for ¡ (the usual recommendation is to access it by !\`), and > for ¿ (usually ?\`). This smacks of the shortcut ‘it works!’ approach which always makes me nervous. He also notes the use of | to obtain an em-dash! I begin to despair. This way lies madness. And why would you want a ‘lone’ tilde character or an underscore in your document? Yes I know they are widely used in Unix for paths in directory and file names, but that’s hardly mainstream T_EX usage. I would be wary of defining a command \big in plain T_EX (as on page 100), since it could easily confuse maths typesetting (try \big\lbrace after you have redefined \big). And it must be incorrect to maintain (page 21) that if a document contains ‘no forward references then it can be formatted in one pass’. If it contains no references at all, yes, this is true, but if there are any references (i.e. \label commands) the aux file has to be written out fully and read again on the next run to resolve the reference(s). In reality this is hardly a problem. I’ve never yet been able to write a document in one ‘pass’. There is always some other reason for having to run L^AT_EX again. Where it does matter is with a document you either receive, or one you pick up from a server. You have to know to run it *at least* twice, and perhaps more times. Basically, keep running it until L^AT_EX stops carping.

To me a section entitled, ‘T_EX for beginners’ contains a bizarre, almost oxymoronic, concept. I no longer believe that you should be taught T_EX, until you have mastered L^AT_EX. If you need to go further than L^AT_EX, then it is worthwhile finding out how T_EX works. Most of us don’t really need to know first about macros and registers, far less ‘token lists’ (unless of course we already have a few computer languages to our credit). There also seems to be some confusion in describing what T_EX is. Norm writes “T_EX allows you to separate markup and layout. Logical divisions in the text (chapters, sections, itemized lists, etc.) are identified by control sequences”. Yes this is true, but it is closer to a description of L^AT_EX. On page 14 we have a supposed comparison of T_EX and troff commands, except the T_EX is L^AT_EX. In passing, why troff? Surely no-one would select troff over L^AT_EX given a free choice. Ten or fifteen years ago, and tied to Unix, maybe there was a contest, but today? Similarly the T_EX macros on page 43 are for L^AT_EX (to be fair, he does say that they are from the CTAN directory macros/latex/contrib/misc so I should have been able to work that one out by myself. If Norm’s contention was that to talk of T_EX was to imply L^AT_EX, I wouldn’t mind, but there is just too much jumble here, and I’m getting confused. Am I just quibbling? I don’t think so. The seeds of confusion are being sown.

Having said all this, I think the book is very useful, and it contains answers to many questions. Just ignore the bits about T_EX and concentrate on the tools. It almost answered my query about PFM files and how to convert them to t_fms. Simply by recording where many things are in the CTAN archives is a great boon. I can happily spend all day searching through CTAN, but my chances of coming across something useful are slim. Now I have a better idea of what is there and my searching is more directed. In some areas Norm does go into reasonable detail – for example the installation of emT_EX, or describing how to use new PostScript fonts in T_EX. He also usefully spends a little time and effort describing how to get pictures into T_EX, describing the problems and pitfalls, as well as actually doing it. It is a useful addition to my library of T_EX books, if only because it more easily enables me to answer those ‘is there a public domain driver for xxx?’ questions.

I’m not sure I enjoy the format very much. Paragraphs without indentation and a paragraph separation of about one

line height make me shudder a bit, but many manuals have this sort of form, and maybe that's what Norm wanted. I don't think it shows T_EX off to advantage, although he did make a real effort by eschewing Computer Modern and using Garamond instead.

X Malcolm's Gleanings

Malcolm Clark
m.clark@warwick.ac.uk

1 Whom the gods would destroy...

At a recent (networking) conference I received a pamphlet about an organisation named Dante (Delivery of Advanced Networking Technology to Europe). What or whom do you associate with this name? Naturally I think first of Dante Alighieri and his romance with Beatrice, then the firm who presses my favourite extra virgin olive oil, and lastly our German sister T_EX organisation. I was therefore fascinated to hear a bit of unsupported scurrilous gossip which maintained that the German DANTE were challenging this new network organisation to change its name — presumably to avoid any possible confusion. This story is probably untrue. Another piece of unsupported news was that DANTE was reluctant to support Haralambous' and Plaiice's Omega project since it was thought to compete with NTS. Such a sad and blinkered view is clearly not worthy of the largest T_EX organisation in Europe, which boasts openly of its financial health. This gossip must therefore be malicious and/or mischievous. Expunge it from your minds!

2 Garnered gleanings

In a flier headed *Quality without compromise* the Royal Society announces its intention of using T_EX to handle papers for *Proceedings: Mathematical and Physical Sciences*. The plan seems to be to encourage authors to submit T_EX files on disk. Wonder why they don't get an email link? A positive intention is to reduce publication times, with a 'fast stream' time of only 15 months. I'm sure we wish them luck.

And in the Royal Statistical Society's *News* the virtues of the 4T_EX CD is extolled, and is given a 'best buy' rating. Allan Reese (for it was he) omits to reveal how one might obtain this gem (albeit flawed), and more important, misses an opportunity to bring this group to the attention of those purveyors of lies and damn lies.

3 Chimes at midnight

I have a weakness for the late Orson Welles. I doubt if there is any of his work that I can say that I didn't enjoy to some (positive) extent. Even his last rambling and chaotic film, *Don Quixote*, has moments of poetry and grace which provide echoes with his youthful output. There is an obvious identification between Quixote and Welles. The film was a labour of love which spanned something like twenty years, and absorbed the proceeds of less 'honourable' projects, like those sherry ads. In the end, it was unfinished, and it was his widow(?) who completed the film. To adopt the mantle of the venerable Don is a perilous undertaking, and perhaps the relative lack of success that Welles enjoyed in the latter part of his career was a response to this identification. What then can we expect for another who has adopted this same mantle, Don Hosek? Don, or as he is now, D A Hosek, is the prime mover of *Quixote Digital Typography* (I suspect he is the entirety of QDT...). The latest venture of QDT is *Serif* (sub-titled *The Magazine of Type & Typography*). This is a quarterly magazine, and so far I have seen only the inaugural issue which was published late in 1994.

What is *Serif* about? I've lost the prospectus (which was interesting enough to encourage me to part with some money for a subscription), so we depend on the contents to guide us. Oddly there is no manifesto or statement of aims in the magazine. The contents include discussions of typefaces, analogue and digital, a taxonomy of letterforms, examination of the output of some contemporary foundries, book reviews... On the whole quite varied and potentially different. QDT has managed to acquire a number of quite notable type-people to contribute: Robert Bringhurst (who I hadn't heard of as a typographer, despite an excellent pedigree, but who I recall as a poet), Charles Bigelow (probably known to T_EXies under the guise of 'Bigelow and Holmes', joint creators of Lucida), Gunnar Swanson (a frequent contributor to the `typo-1` list server), and of course, Don Hosek himself.

I have to admit I found the contributions hard to thole. There was an air of introspective, insider exclusivity which I found frankly irritating. The whole thing came across as either precious or precocious. Take for example the insistence of describing Trump Medieval as Trump Mediëval. In German it is often necessary to indicate a 'missing' 'a' with a dieresis or umlaut. It isn't needed in English where both medieval and mediaeval are acceptable (according to the

OED). Nor is this a form used by either Linotype or Monotype in their catalogues. So why use it here, unless as a form of snobbery? Maybe I'm just irredeemably ignorant. So much read like Irene Handl's famous send up of a Critics' Forum, although I would change her quote from 'there was a certain ragged earnestness, combined with a subtlety, about it, which I found irresistible' to 'entirely resistible', and I'm uncertain about the subtlety. Maybe that's it: maybe it *is* a send up — but how elaborate. No, I suspect that we are being expected to take it earnestly.

Probably there is much good stuff here (although I'd take issue with describing photography as 'modern' in 1890: the daguerreotype dates from 1839, and the glass negative from 1851), but it seems designed to impress rather than to inform. But who is it intended to impress? It can't be those with merely a passing interest in type: it must be those devoured by the subject — are there enough of them, and will they read this? Typographers (in a wide sense) are just as pig-headed and opinionated as the rest of us, perhaps even more so, and one could easily see factional in-fighting developing between them.

An interesting feature of the magazine is that it is set entirely in T_EX. Don did discuss and demonstrate some of the techniques he used to achieve this at last year's TUG conference in Santa Barbara (a most excellent talk). But: I do notice a few infelicities — not to do with T_EX particularly, but to do with proof reading. There is one T_EX failing however: there are far too many hyphens. Three successive hyphens are not uncommon, but the maximum number in a row (or at the end of a row) was at least 5. This occurs in a paragraph of 23 lines, where there are, in total, 10 hyphens. Let's be entirely fair and note that the magazine is set double column, with a measure of about 35–38 characters. That is fairly short: we are talking about 8 words to the line. Hyphens can be tweaked by appropriate loading of the right parameters, but I think this demonstrates that T_EX's line breaking algorithm is not actually as good as we claim it is. Working with lines of the length that Knuth uses in the T_EXbook, it is possible to minimise hyphens and to ensure that successive hyphens are rare. Working with narrow measure, T_EX all too often throws up its hands and rolls over on its back. Still on hyphenation, I found coor-dinates rather unexpected. There are also a couple of widow lines which could have been avoided, but this I suspect has more to do with proofing than T_EX. Otherwise we have both Morris' and Morris's, 'teh' for 'the' (not uncommon, but even a spell checker finds this), Updikes' (who was this man Updikes? is he related to Updike?) and the odd 'e nd' (again easy to find with a spell checker).

I'll be interested to see what the next few issues bring, but right at this moment I'm not too sanguine. Maybe if I knew why the issue was dedicated to St Therese of Lisieux (no accents this time) I would be enlightened. I just hope we shouldn't be lighting candles to St Jude.

4 Message in a bottle

TUGboat 15(2) arrived some time before Christmas, looking thin and wan at about 70 pages. I have bemoaned the late arrival of *TUGboat* in this column before. Daily I await the other two promised editions for 1994. One will be the conference proceedings, but I suspect the other normal issue will also be thin and weedy. I tried raising the issue of the timeliness of *TUGboat* at the TUG AGM at Aston in 1993, to have the discussion turned round by Phil Taylor who viewed any criticism of *TUGboat* as a criticism of its editor, the inestimable Barbara Beeton. Better to have it high quality but late I was told, and the audience applauded. I despair. If T_EX is a production quality tool then we have to explain the non-appearance of *TUGboat* very carefully. A few years ago, when *TUGboat* was similarly late, rumours started to circulate that TUG itself had gone out of existence. This warning was obviously not enough. Perhaps the changes in editorial policy noted in this issue will eventually overcome the scheduling difficulties.

There were a few gems however: the report of the NTS (New Typesetting System) comprises ten (canonical?) points. One discusses the logo they should use. As I think Thora Hird used to say, 'get the shoes right, and everything else falls into place'. Sebastian was right: 'a pox on a logos'. And apparently the L^AT_EX3 project team has decided that L^AT_EX 2_ε is L^AT_EX, and L^AT_EX is L^AT_EX209. I'm reminded of the venerable Leslie Lamport's wisdom on the pronunciation of L^AT_EX (or was that L^AT_EX209?) '... best determined by usage, not fiat'. Continuing a fairly recent tradition, *TUGboat* also contains abstracts of other T_EX journals, including these very same Annals. I find these *Gleanings* occupying an inordinate amount of space, perhaps because it is easier to summarise trivia than substance.

5 What next?

Since TUG moved its offices to Santa Barbara there have been earthquakes and floods. The Los Angeles earthquake of 94 didn't do much damage locally — only a few more cracks in the building TUG inhabits. The floods earlier this year did affect SB quite badly, although again the TUG office survived. Maybe the Tugboat notion wasn't so bad after all. As for plague — well, phylloxera is endemic in the Napa Valley. Were I in philosophical mood, I might see these events as an allegory on TUG itself.

XI Report of the 1994 UKTUG AGM

R. A. Bailey
Hon. Secretary, 1991–95

Official report of the AGM of the UK T_EX Users Group, held at the University of Warwick (The Staff Club Quiet Room, Rootes Building)

on Wednesday 19 October 1994 at 1100 hours

There were about twenty members present. The following is a brief summary of the business transacted; it is categorized by, roughly, the numbered agenda items.

1. Report of the 1993 AGM

This report had already been published in *Baskerville* 3.2. Copies were also available at the meeting. The report was received as correct.

2. Chairman's Report

The Chairman, C. A. Rowley, made his report, which is printed following this article.

3. Approval of Accounts

The Treasurer, P. Abbott, gave his report. Copies of the unaudited accounts for 1993–94 were presented. When audited, the accounts will appear in *Baskerville*. A report on membership appears separately.

The following points were made during the discussion of the report.

- Although the financial year runs from 1 August to 31 July, these accounts covered the period 31 August 1993 to 31 July 1994 only, because the accounts had been closed late the previous year.
- The meeting on 1 May 1993 had been a visit to John Wiley and Sons, at Chichester, arranged by G. Granger. Because of the generosity of Wiley, the Group had incurred no expenditure on this meeting. The meeting thanked John Wiley and Sons for its generosity.
- Although TUG '93 had been run by a separate committee, it had used the UK T_EX Users' Group bank account to save on bank charges. Therefore the accounts show some items connected with TUG '93 which do not formally form part of UKTUG's activities: there should be no such items in future accounts.

4. Appointment of Auditor(s)

The Treasurer reported that Colin Smith would audit the accounts for 1993–94. Colin Smith was reappointed auditor for 1995.

5. Membership Fees

The Treasurer proposed the following motion, on behalf of the committee:

The membership fee for 1995 shall be £20.00 for full membership or £10.00 for full-time student membership. Anyone who joins the UK T_EX Users' Group on or after 1 October 1994 will, for the above fee, become a member until 31 December 1995, although they will be ineligible to vote on business of the UK T_EX Users' Group until 1 January 1995.

P. Abbott, in his personal capacity, then proposed an amendment: that the second sentence be replaced by

Any individual member who pays his or her subscription for one calendar year before the end of the previous calendar year shall be entitled to pay at the rate for the previous calendar year.

R. Fairbairns seconded the amendment, which was passed *nem. con.* The amended motion was then passed *nem. con.*

6. Meetings Fees

The Treasurer proposed the following motion, on behalf of the committee:

Meetings and workshops shall be costed to break even. The surcharge for non-members shall be the greater of £10 and rounded 10% of the members' meeting fee. There shall be a discount for booking before a specified date.

J. Fine then proposed an amendment: that the word *ordinarily* be inserted after the second occurrence of the word *shall*. M. Clark seconded the amendment, which was passed *nem. con.* The amended motion was then passed *nem. con.*

7. Institutional Membership

reprinted from Baskerville

Volume 5, Number 1

The Committee Secretary reported that, in accordance with Clauses 2(1) and 15(2) of the Constitution, the Committee had determined that the regulations for institutional members should be:

1. *Each institutional member shall nominate a single person as a point of contact.*
2. *The person so nominated shall receive five copies of each issue of Baskerville.*
3. *For each institutional member, up to 20 people from that institution shall be allowed to attend UK T_EX Users' Group meetings at the members' rate, so long as they identify themselves, on the booking form, as being among their institution's quota.*

There was some discussion of this issue, with members pointing out that the regulations should specify the voting rights, if any, and eligibility for Committee membership, of persons belonging to an institutional member. It was agreed that the Committee would resolve these issues, and publish revised regulations early enough that Institutional Membership would be available for 1995. The revised regulations are published elsewhere in this issue of *Baskerville*.

8. The institutional members annual subscription

The Treasurer proposed the following motion, on behalf of the committee:

The membership fee for 1995 shall be £100.00 for institutional membership.

This was passed *nem. con* with one abstention. However, the Committee was asked to consider whether, in future, the quoted sum of £100 could be replaced by a phrase such as *five times the annual subscription for a full member* so that there would not need to be a separate vote on the level of the subscription for institutional members.

9. Election of Committee

Of the previous committee (excluding the Chair), P. Abbott (as Treasurer), M. Clark, J. Fine and A. S. Jeffrey continued. Of those retiring, R. Fairbairns stood for re-election. Further nominations for committee membership had been received for D.P. Carlisle and C. Hewlett. These three people were all elected to the committee, bringing the total size of the committee to seven (excluding the Chair).

Subsequently, the committee co-opted R. A. Bailey and S. P. Q. Rahtz.

10. Report on *Baskerville*

S. P. Q. Rahtz, the current editor the Group's newsletter *Baskerville*, gave a short report on the first full year in which *Baskerville* had appeared at intervals of approximately two months. He thanked the members of the Committee for their help and support, and all contributors. He encouraged all members to send in articles for publication in *Baskerville*.

11. Topics for Meetings

The membership made the following suggestions for meetings.

- Basic training. (This idea provoked much discussion, because although there always seems to be a demand for basic training, those training courses that are organized by the UK T_EX Users' Group do not attract many participants. It was suggested that perhaps the Group should simply provide trainers at meetings organized by others, or pass on names of professional (La)T_EX trainers.)
- A meeting for university academic registrars, or members of their departments, to address such issues as: regulations for the appearance of PhD theses set in (La)T_EX; databases for syllabuses of modules in modular degrees.
- Front ends to T_EX, with demonstrations and stalls where participants could try out the front ends on PCs.
- Conversion between word-processing packages and T_EX.
- Organization by the UK T_EX Users' Group of part of the next EuroT_EX meeting, which will be in Maastricht in 1995.

12. The Archive

R. Fairbairns reported that the 'Aston Archive' had been successfully moved to Cambridge, where it was now housed at the University of Cambridge Computer Laboratory. It was being run on a shoestring, but its home was secure. It was being run by the volunteer labour of R. Fairbairns, M.A. Johnson and S. P. Q. Rahtz. The meeting expressed its sincere thanks to R.M. Needham, the head of the University of Cambridge Computer Centre, for allowing the Archive to be mounted there.

13. Services to Members

P. Abbott and R. Fairbairns reported that the Committee had increased its range of services to members during the year (for more details, see the Chair's report). In addition, a special issue of *Baskerville* devoted to 100 Frequently Asked Questions would be published in December 1994.

P. Abbott asked if the Group should have a fax number. The Committee said that it would look into the suggestion.

14. Thanks

P. Abbott thanked the retiring members of the Committee for their work. P. Taylor thanked P. Abbott for creating the Archive at Aston and maintaining it there for so many years.

The meeting closed at about 1300 hours.

XII Membership notes

Peter Abbott

1 Institutional Membership

At its meeting in November 1994, the Committee of UKTUG decided upon the following regulations for institutional membership of the UK T_EX Users' Group, with effect from the beginning of 1995.

1. Each institutional member shall give the UK T_EX Users' Group its name in its preferred form for inclusion in a list of institutional members.
2. Each institutional member shall nominate a single person as *contact person*, and inform the UK T_EX Users' Group of that person's address.
3. Each institutional member shall nominate up to five people from that institution to be deemed individual members of the UK T_EX Users' Group for the purposes of eligibility to serve on the committee and eligibility to vote. The contact person may, but need not, be one of these nominated people.
4. The designated contact person shall receive five copies of all written material sent to individual members of the UK T_EX Users' Group, including each issue of *Baskerville*, information about forthcoming meetings and information about elections. It is the responsibility of the contact person to forward this material to the nominated people.
5. For each institutional member, up to 20 people from that institution shall be allowed to attend UK T_EX Users' Group meetings at the members' rate, so long as they identify themselves, on the booking form, as being among their institution's quota.

2 UKTUG membership report

I would like to thank all those members who responded promptly in renewing membership for 1995. The 1995 membership (as at 16th January 1995) is:

UKTUG	42
UKTUG (student)	1
UKTUG and full TUG	67
UKTUG and basic TUG	4
UKTUG and full TUG (student)	4
UKTUG and basic TUG (student)	0
UKTUG Institutional	2
full TUG	2
basic TUG	0
full TUG (student)	0
basic TUG (student)	0
<hr/> Total	<hr/> 122

I have replied to every renewal, either by email, fax or paper mail so if you have paid and not received an acknowledgement please contact me. This will be the last issue of *Baskerville* that 1994 members will receive and I would appreciate details as to why members are not renewing. Elsewhere you will read a letter from a member who decided not to renew and we trust that the points raised have been covered. In this way we hope to continue to improve the service you get from the Group.

XIII Chair's report on 1993–94

Chris Rowley

First, a big thank you to all my colleagues on the Committee for all their hard work during the last year.

The notable innovation in 1994 has been the regular appearance of *Baskerville*. Under the editorship of Sebastian Rahtz, and with Robin Fairbairns and Jonathan Fine as publisher and distributor, issues of our newsletter now reach members approximately every two months. It contains an interesting variety of T_EX-related articles in addition to notices of meetings and subscription forms. In my opinion, *Baskerville* is the best in content and, by a wide margin, in timeliness of all the journals and newsletters produced by T_EX user groups. Thanks to Sebastian, Robin and Jonathan, and also to all those who have written items for *Baskerville*.

Four meetings were held during the year. A meeting on 'Front Ends to T_EX' was held at Aston University on 20 October 1993, after the AGM. This was organized by Sebastian Rahtz and Peter Abbott, and was attended by about 30 people. Talks and demonstrations were given by Adrian Clark, Nikos Drakos, Jonathan Fine, Christopher Mabb and Sebastian Rahtz.

On 18 January 1994 a meeting was held at Rewley House in Oxford on the theme 'Choosing and Using PostScript Fonts with T_EX'. It was organized by Alan Jeffrey, Sebastian Rahtz and Ian Hall, and was attended by about 30 people. Talks were given by Angus Duggan, Alan Jeffrey, Sebastian Rahtz and Will Shaman, and the meeting concluded with a panel session.

The big meeting of the year was the two-day meeting at the University of Warwick: 'L^AT_EX 2_ε: the conference', on 21–22 March 1994. It was organized by Malcolm Clark and me. The talks by members of the L^AT_EX3 Project Team (Johannes Braams, David Carlisle, Alan Jeffrey, Frank Mittelbach and me) and Michel Goossens introduced the audience of 80 to the philosophy and features of the new standard version of L^AT_EX, which had been released in beta-test in December 1993 and whose first full release was in June 1994.

On 11 July 1994 a training meeting 'L^AT_EX fonts and graphics: a hands-on tutorial' was held in Cambridge. The organizers were Robin Fairbairns and Jonathan Fine; the trainer was Alan Jeffrey, and extra training material was provided by David Carlisle. The 13 participants found the meeting very worthwhile: it was a pity that no more people wanted to attend.

Thanks to all the organizers, speakers and trainers for making these meetings possible. Especial thanks to Malcolm Clark, who so successfully was the local organizer for the residential meeting.

Other members of the Committee may have been less visible, but have been doing sterling work in the background. For example, Peter Abbott is to be congratulated on sorting out the Group's accounts and financial position. Peter has also increased the range of services available to members, who can now buy, in addition to a range of books, disc sets of emT_EX and OzT_EX at cost price, and the CDROM of T_EXware produced by the NTG.

The 'Aston Archive' has hitherto been strictly separated from the UK T_EX Users' Group. However, during the year it was brought to the Committee's attention that the Archive might not be able to continue at Aston University. The Committee felt that the Archive is a considerable service to members of the UKTUG, and so took upon itself the investigation of another suitable home for it. After the investigation of several possible homes, a smooth transfer of the Archive to Cambridge University has now been achieved. Thanks to Roger Needham at Cambridge for authorizing this; and to Robin Fairbairns, Martyn Johnson and Sebastian Rahtz for organizing the changeover and maintaining this UK node of the CTAN.

A brief history of the relationship between T_EX and Aston university has already appeared in these annals but I should like to formally record here the enormous debt of gratitude owed by this group and the world-wide T_EX community to Peter Abbott for organising the large range of activities that made Aston the centre of the T_EX universe for many years.

Finally, I should say something about TUG and our relationship with it. As a result of discussions at the TUG meeting at Santa Barbara in July 1994, the relationship between TUG and the various other user groups has come under scrutiny. It is possible that the role of TUG will change. There were further discussions at the EuroT_EX meeting in Gdańsk in September 1994, but the picture is still far from clear. I will keep members of UKTUG informed of developments.

XIV The UK T_EX Users Group

1 Group aims and activities

The UKTUG was founded in 1989; its activities and benefits include:

- Regular meetings covering both T_EXnical matters and general typographic subjects;
- Visits to publishers, font suppliers and book producers;
- In-depth workshops, with hands-on computer sessions where appropriate;
- Negotiation of discounts on T_EX relevant books with publishers.
- Distribution of public domain T_EX implementations (in cooperation with the international T_EX archives);
- Discount on joint UK T_EX Users Group/TUG membership;
- Reduction in conference fees for the TUG annual meeting and the annual European meetings;
- The publication and distribution of *Baskerville* (the annals of the UK T_EX Users Group).

2 Programme of meetings for 1995

July, 1995 (T_EX Users Group Annual Meeting in Florida)

September, 1995: emT_EX in depth: a training meeting on MSDOS T_EX, with Sebastian Rahtz Location: Oxford.

September, 1995 (EuroT_EX meeting in the Netherlands)

October 1995: AGM, at Queen Mary and Westfield College, University of London, with sessions on ‘non-traditional’ T_EX.

All members of UKTUG will be provided with more details of these meetings as they become available. Reports of the meetings will appear in *Baskerville*, the organ of the group.

The 1994–95 UKTUG committee

C. A. Rowley Chair
P. Abbott Treasurer and
 Membership Secretary
R. A. Bailey Committee Secretary
M. Clark Meetings Secretary
S. P. Q. Rahtz *Baskerville* editor
D. P. Carlisle; R. Fairbairns; C. Hewlett; A. S. A. Jeffrey.

Book Discounts for UKTUG members

The UK T_EX User group has been in discussion with a number of publishers to arrange discounts on publications which may be of interest to members. We have arrangements with Addison-Wesley for their well-known T_EX-related publications, and with International Thomson Publishing to supply any of the very excellent O’Reilly & Associates Inc. series of books to members.

The agreed list of books, together with the discounted (at least 10%) price, is distributed occasionally with *Baskerville*, but is always available from the Treasurer, Peter Abbott. The quoted price includes the cost of postage and packing.

We would like to extend this service to members. If you have contacts with the sales and marketing parts of any publisher who publishes relevant books, please inform us and we will try to arrange something mutually convenient and beneficial.

We are only allowed to offer this service to **current** members of the UK T_EX User Group and/or members of TUG. Please send your order and cheque (in UK £) to Peter Abbott (address in *Baskerville* masthead). Make cheques payable to ‘UKTUG’ please. Books from Addison-Wesley are delivered direct but books from O’Reilly will be routed through UKTUG. *In all cases* please notify Peter Abbott by email, phone, fax or letter when books are delivered.

XV Obtaining T_EX

edited by Sebastian Rahtz

From the network

The UK T_EX Archive on `ftp.tex.ac.uk` is part of the CTAN (Comprehensive T_EX Archive Network) collaborating network of archives on the Internet organised by the T_EX Users Group. The three main archives follow the same structure and have identical files (`ftp.tex.ac.uk`, `ftp.shsu.edu` and `ftp.dante.de`).

The CTAN archives all run an enhanced *ftp* server which supports dynamic compression, uncompression, and archive creation options. Fetch the top-level file `README.archive-features` for information. The server also supports site-defined commands to assist you. Please read `README.site-commands` for a brief overview.

Details of where to find public domain, or shareware, T_EX packages for different machines and operating systems are given below.

Please report any problems with CTAN archives via email to `ctan@shsu.edu`. The entire archive is available on CDROM as the 'T_EXcetera' package from Prime Time Freeware, 370 Altair Way, Suite 150, Sunnyvale, CA 94086 USA (`ptf@cfc1.com`). This is also distributed by the TUG office, and in the UK by Lasermoon Ltd (email `info@lasermoon.co.uk`, phone 01329 826444).

Unix tapes

For a complete Unix T_EX distribution, a 1/4 inch cartridge, QIC-120 or QIC-150 format (DC600A or DC6150) can sent with envelope *and* stamps for return postage to:

David Osborne
Cripps Computing Centre,
University of Nottingham,
Nottingham NG7 2RD

Due to currency exchange, this service is offered only within the UK.

PC and Mac disks

The UKTUG distributes an emT_EX kit for PCs, and an OzT_EX kit for Macintosh. The cost covers copying and postage costs, and the shareware fee for OzT_EX. Each set costs £30, and is available from Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Cheques must be payable to 'UKTUG'. Please note that this service *is available to UKTUG members only*. Each set comes with an installation guide, and (at least) full T_EX and METAFONT, a previewer, a PostScript driver, and CM fonts. Two update disks a year will be sent out automatically, with the current version of L^AT_EX 2_ε, and other goodies. A subscription service will be available for subsequent years. In addition, subscribers can request up to 3 disks a year with any material from the CTAN archives, but this will be supplied 'as is', without instructions.

Enquiries for T_EX for the Atari ST etc. can be directed to: The Fast Club, 7 Musters Road, Nottingham NG2 7PP. Phone 01602 455250, fax 01602 455305. They also supply a variety of T_EX-related software in Atari format.

reprinted from Baskerville

Volume 5, Number 1

Name	Environment	CTAN path	Notes
em \TeX	DOS, OS/2	systems/msdos/emtex	the betatest drivers, and 386-specific versions of \TeX and MF, are in the <code>betatest</code> subdirectory
sb \TeX	DOS	systems/msdos/sbtex	includes latest \TeX and METAFONT
g \TeX	DOS	systems/msdos/gtex	386-only \TeX and METAFONT, which replace parts of em \TeX , and work with Windows memory management
Oz \TeX	Mac	systems/mac/oztex	This package is shareware
CMac \TeX	Mac	systems/mac/cmactex	Shareware port of Unix \TeX
Direct \TeX	Mac	systems/mac/directtex	This package is shareware
pas \TeX	Amiga	systems/amiga/pastex	
—	Atari	systems/atari/lindner-tex-packed-disks	
—	Atari	systems/atari/cs-tex	
web2c \TeX	Unix	systems/unix/web2c	a complete source kit for \TeX and METAFONT, which should compile on most Unix boxes; needs a C compiler.
—	VMS	systems/vms	Alpha and VAX versions available

Book list

Addison-Wesley Books					
The T _E X book	Donald E Knuth	0 201 13448 9	paper	20.50	
The T _E X book	Donald E Knuth	0 201 13447 0	hard	29.50	
T _E X the program	Donald E Knuth	0 201 13437 3	hard	34.00	
The METAFONTbook	Donald E Knuth	0 201 13444 6	paper	18.50	
The METAFONTbook	Donald E Knuth	0 201 13445 4	hard	29.50	
METAFONT the program	Donald E Knuth	0 201 13438 1	hard	34.00	
Computer Modern Typefaces	Donald E Knuth	0 201 13446 2	hard	34.00	
3:16 Bible texts illuminated	Donald E Knuth	0 895 79252 4	paper	19.50	
The CWEB system of structured documentation	Donald E Knuth & Silvio Levy	0 201 57569 8	paper	17.00	
L ^A T _E X: a document preparation system, second edition	Leslie Lamport	0 201 52983 1	paper	24.00	
The L ^A T _E X Companion	Michael Goossens, Frank Mittelbach & Alexander Samarin	0 201 54199 8	paper	24.00	
Introduction to T _E X	Norbert Schwarz	0 201 51141 X	paper	19.50	
A guide to L ^A T _E X	Helmut Kopka & Patrick Daly	0 201 56889 6	paper	22.00	
T _E X by topic	Victor Eijkhout	0 201 56882 9	paper	22.00	
T _E X for the impatient	Paul W Abrahams, Kathryn Hargreaves & Karl Berry	0 201 51375 7	paper	20.50	
Postscript language program manual	Adobe	0 201 18127 4	paper	24.00	
Postscript language, tutorial & cookbook	Adobe	0 201 10179 3	paper	15.00	
Portable Document Format Reference Manual	Adobe	0 201 62628 4	paper	18.50	
The Postscript Font handbook	Jonathon Grosvenor, Kay Morrison & Alexander Pim	0 201 56893 4	paper	22.00	
Postscript by example	Henry McGilton & Mary Campione	0 201 63228 4	paper	23.00	
Thinking in Postscript	Glenn Reid	0 201 52372 8	paper	18.50	
Collier's rules for desktop design and typography	David Collier	0 201 54416 4	paper	15.00	
Interactive Typography	David Collier & Ian Martin	0 201 63167 9	cd-rom	tba	
O'Reilly Books					
4.4BSD-Lite CD-ROM Companion		1 56592 092 9		26.55	
4.4BSD System Manager's Manual		1 56592 080 5		19.80	
4.4BSD User's Reference Manual		1 56592 075 9		19.80	
4.4BSD User's Supplementary Documents		1 56592 076 7		19.80	
4.4BSD Programmer's Reference Manual		1 56592 078 3		19.80	
4.4BSD Programmer's Supplementary Documents		1 56592 079 1		19.80	
Notable Speeches of the Information Age (tape)	Barlow	1 56592 992 6		6.75	
Power Programming with RPC	Bloomer	0 937175 77 3		19.80	
The Frame Handbook	Branagan, Sierra	1 56592 009 0		26.55	
Learning GNU Emacs	Cameron	0 937175 84 6		18.45	
ORACLE Performance Tuning	Corrigan	1 56592 048 1		32.40	
Sendmail	Costales	1 56592 056 2		22.00	
Using C on the UNIX System	Curry	0 937175 23 4		16.65	
SCO UNIX in a Nutshell	Cutler	1 56592 037 6		6.75	
Checking C Programs with lint	Darwin	0 937175 30 7		8.55	
sed / awk	Dougherty	0 937175 59 5		18.45	
High Performance Computing	Dowd	1 56592 032 5		17.35	
Software Portability with imake	DuBois	1 56592 055 4		16.65	
Connecting to the Internet	Estrada	1 56592 061 9		10.75	
!%@::A Directory of Electronic Mail Addressing / Networks 4e	Frey	1 56592 046 5		6.75	
Essential System Administration	Frisch	0 937175 80 3		19.80	
POSIX.4: Programming for the Real World	Gallmeister	1 56592 074 0		19.76	
PGP: Pretty Good Privacy	Garfinkel	1 56592 098 8		11.95	
Practical UNIX Security	Garfinkel	0 937175 72 2		19.80	
UNIX in a Nutshell- System V /Solaris 2.0 2e	Gilly	1 56592 001 5		6.75	
Understanding and Using COFF	Gircys	0 937175 31 5		14.65	
Using Netnews	Harrison	1 56592 101 1		16.65	

TCP/IP Network Administration	Hunt	0 937175 82 X	19.80
Multi-Platform Code Management	Jameson	1 56592 059 7	26.55
Migrating to Fortran 90	Kerrigan	1 56592 049 X	16.65
Linux Network Administrator's Guide	Kirch	1 56592 087 2	16.65
The Mosaic Handbook for the Macintosh	Koman	1 56592 096 1	19.80
The Mosaic Handbook for Microsoft Windows	Koman	1 56592 094 5	19.80
The Mosaic Handbook for the X Window System	Koman, Ferguson	1 56592 095 3	19.80
The Whole Internet Users Guide /Catalog 2e	Krol	1 56592 063 5	16.65
Learning the vi Editor 5e	Lamb	0 937175 67 6	14.65
lex /yacc 2e	Levine	1 56592 000 7	19.80
POSIX Programmer's Guide	Lewine	0 937175 73 0	23.20
Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs	Libes	1 56592 090 2	19.75
DNS / BIND	Liu	1 56592 010 4	19.80
Managing Internet Information Services	Liu, Peek, Jone, Buus, Nye	1 56592 062 7	19.75
System Performance Tuning	Loukides	0 937175 60 9	16.65
UNIX for FORTRAN Programmers	Loukides	0 937175 51 X	16.65
Understanding Japanese Information Processing	Lunde	1 56592 043 0	20.65
Security and Networks (tape)	Malamud	1 56592 997 7	6.75
Networked Information and Online Libraries (tape)	Malamud	1 56592 998 5	6.75
European Networking (tape)	Malamud	1 56592 999 3	6.75
Global Network Operations (tape)	Malamud	1 56592 993 4	11.25
Mobile IP Networking (tape)	Malamud	1 56592 994 2	6.75
The Future of the Internet Protocol (tape)	Malamud	1 56592 996 9	16.65
Encyclopedia of Graphics File Formats	Murray, vanRyper	1 56592 058 9	39.60
Managing Projects with make 2e	Oram	0 937175 90 0	11.95
Practical C Programming 2e	Oualline	1 56592 035 X	16.65
MH /xmh: E-Mail for Users/Programmers 2e	Peek	1 56592 027 9	19.80
UNIX Power Tools with CD-ROM	Peek	0 553 35402 7	39.60
Love Your Job!	Powers	1 56592 036 8	8.55
Building a Successful Software Business	Radin	1 56592 064 3	16.65
Understanding DCE	Rosenberry	1 56592 005 8	16.65
Distributing Applications Across DCE and Windows NT	Rosenberry	1 56592 047 3	16.65
Learning the Korn Shell	Rosenblatt	1 56592 054 6	18.45
Computer Security Basics	Russell	0 937175 71 4	19.80
Smileys	Sanderson	1 56592 041 4	4.00
Learning Perl	Schwartz	1 56592 042 2	16.65
guide to Writing DCE Applications 2e	Shirley	1 56592 045 7	22.50
Managing NFS and NIS	Stern	0 937175 75 7	18.45
Programming with curses	Strang	0 937175 02 1	8.55
termcap /terminfo	Strang	0 937175 22 6	14.65
Manging UUCP and Usenet 10e	Todino	0 937175 93 5	18.45
Using UUCP and Usenet	Todino	0 937175 10 2	14.65
Learning the UNIX Operating System 3e	Todino	1 56592 060 0	6.10
Programming Perl	Wall	0 937175 64 1	19.80
Making TeX Work	Walsh	1 56592 051 1	19.75
<i>The X Window System Series</i>			
The X Window System in a Nutshell	X Window System, Release 6, Companion CD	1 56592 084 8	19.75
X Protocol Reference Manual Vol 0	Nye	1 56592 008 2	23.20
Xlib Programming Manual Vol 1	Nye	1 56592 002 3	23.20
Xlib Reference Manual Vol2	Nye	1 56592 006 6	23.20
X Window System User's Guide Vol 3	Quercia	1 56592 014 7	23.20
X Window system User's Guide Vol 3M	Quercia	1 56592 015 5	23.20
X Toolkit Intrinsic Programming Manual Vol 4	Nye	1 56592 003 1	23.20
X Toolkit Intrinsic Programming Manual Vol 4M	Nye	1 56592 013 9	23.20
X Toolkit Intrinsic Reference Manual Vol 5	Flanagan	1 56592 007 4	23.20