

Baskerville

The Annals of the UK T_EX Users' Group
ISSN 1354-5930

Editor: Sebastian Rahtz

Vol. 8 No. 1
January 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 and News	4
1	Why T _E X?	4
2	The 1997 AGM meeting	5
3	T _E X Live 3	6
II	Chairman's Report to the AGM of UKTUG for the period October 1996 to November 1997	7
III	The pdfT _E X user manual	9
1	Introduction	9
2	Implementation details	9
2.1	Compilation	9
2.2	Search paths	10
3	Fonts	10
4	New primitives	11
5	Graphics and color	13
6	Macro packages supporting pdfT _E X	14
7	ε-T _E X V2: a peek into the future	15
8	Introduction	15
9	Ideas which are almost certain to appear in ε-T _E X V2	15
9.1	Increasing T _E X's registers	15
9.2	Improved natural language handling	15
9.3	Arithmetic expressions	16
9.4	Discards are no longer discarded!	16
9.5	Read-write access to parshape	16
9.6	Interrogating the current conditional context	16
9.7	Access to information concerning font-character combinations	16
9.8	Better debugging aids	16
9.9	Subtle change to the semantics of protected	16
9.10	Optimisations	16
9.11	Access to the components of a glue quantity	17
9.12	Improved typographic quality	17
9.13	Improved typographic quality, cont.	17
10	Ideas still under discussion	17
10.1	Can T _E X find this font?	17
10.2	Maths alignments	17

10.3 Typesetting on a grid	17
11 NTS	18
12 The inaugural meeting of TUG India	19
13 First stirrings	19
14 India, and the inauguration	19
15 The first TUG India courses	20
16 . . . and some sightseeing	20
17 Conclusion	21
18 An introduction to LyX	22
19 What is LyX?	22
20 The main Window	22
21 Getting started	23
21.1 Changing the Appearance	24
22 Lists	24
22.1 Lists within lists	24
23 Footnotes and marginal notes	24
24 Figures and tables	24
24.1 The Table Editor	25
24.2 Figures	25
25 Setting Mathematics	25
26 Document styles	26
27 Entering L ^A T _E X Commands	26
28 Interfacing with L ^A T _E X	26
29 Spell Checker	27
30 A Final Word	27
31 Typesetting the L ^A T _E X Logo: a Saga	28
32 The Old L ^A T _E X coding	28
33 The standard L ^A T _E X coding	28
34 My first attempt	29
35 But the standard code doesn't even work with CM-family fonts!	31
36 Conclusion: future work	32
37 The natbib package	33
38 Introduction	33
39 Previous solutions	33
39.1 The natsci.bst style	34
39.2 The apalike.bst style	34
39.3 The newapa style	34
39.4 The Harvard family	34
39.5 The astronomy style	35
39.6 The authordate style	35
40 The natbib system	35
41 Using this package	35
41.1 Basic citation commands	36
41.2 Extended citation commands	37
41.3 Selecting citation punctuation	38
41.4 Priority of style commands	39
41.5 Other formatting options	39
41.6 Automatic Indexing of Citations	40
41.7 HyperT _E X compatibility	40
41.8 Multiple bibliographies in one document	40
41.9 Sorting and compressing numerical citations	41
41.10 Long author list on first citation	41
42 Numerical citations with author-year styles	42
42.1 Selecting numerical mode	42

Editorial and News

42.2 New bibliography styles	42
43 Local configuration	42
44 L ^A T _E X freelance editors sought	43
45 EuroT _E X'98 and TUG 98	44
46 EuroT _E X'98: Saint-Malo, France — March 29th to 31st, 1998	44
46.1 Connection with WEPT	44
46.2 Calendar	44
46.3 Enquiries	45
47 TUG'98: The 1998 T _E X Users Group Conference	47
48 Bursary for UKTUG members to attend T _E X conferences	47
49 Donald Knuth's 60th birthday	48
50 Provisional minutes of the UKTUG AGM, 26 November 1997	50
51 1. Report of the 1996 AGM	50
52 2. Chair's Report	50
53 3. Treasurer's Report	50
54 4. Apointment of Auditors	50
55 5. Setting annual membership fee	50
56 6. Setting annual institutional membership fee	51
57 7. Confirmation of Philip Taylor as Chair of the Group	51
58 8. Election of Committee Members	51
59 9. Report on Baskerville	51
60 10. T _E Xnical services to members	51
61 11. Motions proposed by members	51
61.1 Motion 11a)	51
61.2 Motion 11b)	52
61.3 Motion 11c)	52
61.4 Motion 11d)	52
62 12. The Future of UKTUG Meetings, and possible topics	52
63 13. Any other business	52
64 UKTUG Information	53
65 The 1997–98 UKTUG committee	53
66 Contacting UKTUG	53
67 Baskerville	53
68 Book Discounts for UKTUG members	53
69 Obtaining T _E X from CTAN	53

I Editorial and News

Sebastian Rahtz

It seems a long time since I last wrote a *Baskerville* editorial, but it is no less pleasurable to be able, once again, to share some T_EXery with you. “Why”, however, the keen-eyed reader will ask, “is this *Baskerville* volume 8, and not 7 still?”. For an explanation of the UKTUG committee’s decisions regarding this, I refer you to the report by the outgoing Chairman, Robin Fairbairns, following this editorial. You’ll also find information about other glories the committee has planned for the members.

1 Why T_EX?

Just after the New Year, I had the pleasure of travelling to India for the first meeting of the Indian T_EX Users Group (a report on this trip can be found elsewhere in this *Baskerville*). In the opening session, I tried to give an explanation of why T_EX was still relevant, and it seems appropriate to offer the points I made to UK readers as well.

When Donald Knuth started T_EX, he thought he was undertaking the simple task of writing a typesetting program which he could use to prepare volumes of his monumental series, *The Art of Computer Programming*; in the event, it called for 10 years work, and the creation of not only T_EX, but a companion font-drawing program, METAFONT, and the Computer Modern family of fonts. During the years of development, Knuth made some significant decisions:

- He decided to place the program in the public domain for all to use; not only was this generous in itself, but it prompted many people to try it, confident in the knowledge that they could find out what it was doing, that it had no secrets.
- He worked on the design substantially alone, albeit with a group of graduate students and the help of research grants; this meant that the main T_EX programs have a single architecture, and are to all intents and purposes free of bugs, the work of an individual not a team.
- Knuth listened to feedback, and encouraged it. This led to the creation of the T_EX Users Group in about 1980, which did much to popularize the software.
- He wrote the programs using his WEB system of literate programming, in which the program code and its documentation are inextricably woven. The closely-documented nature of the code has made it possible for others to understand it. He even *published* the source code in a book, an almost unprecedented action.
- He did not commit himself to a specific output device (unlike troff, which did not achieve the same success as T_EX), but separated out the ‘driver’ functionality into external programs, by defining T_EX’s output as a neutral device-independent format.
- He recognized that if the program were to have a long shelf-life, he could not commit himself to the graphical and colour capabilities of the period. A conscious decision was made to *omit* any specific functionality of this kind, and instead to provide an all-purpose `\special` command
- Lastly, Knuth froze T_EX when he felt it was complete enough. This means that it stays forever as a fixed reference point, generating the same output from the same input and exhibiting no known errors. This allows some to build on it with confidence, others to write their documents knowing that they will always be able to process them.

But things have moved on since Knuth wrote T_EX; nowadays the talk is of desktop-publishing, word-processing packages, HTML, the Internet, PDF, SGML — was it an anachronism to start a new T_EX Users Group in India in 1998? I believe it was not, and I have followed Don Knuth’s favourite game of playing with numbers to explain why. Since this is (just) 1998, I have attempted to find 9.8 reasons why T_EX is still relevant today, and why the Indian group will have much to work on. I will go over my 9 reasons as reflections on the way T_EX and its users are *evolving* to meet today’s needs.

- 1 The Internet. T_EX users (well, not all of them, since it makes sense only if you use structured markup) can interact with the Internet in a variety of ways:
 - Converting L^AT_EX into HTML, using `latex2html`, `Tex4ht`, or a variety of other programs.
 - Having L^AT_EX read directly by a browser add-on, of which IBM’s `techexplorer` seems to be the most fully developed.

- Producing Portable Document Format for display using Acrobat Reader inside a Web browser. Using Han The Thanh's \TeX variant, pdf \TeX , or a dvi to PostScript program with Acrobat Distiller, we can make PDF files; even better, we can add macros such as my \LaTeX `hyperref` package, or Hans Hagen's Con \TeX t, and automatically produce PDF of a richness unsurpassed by almost any other program.
 - There is also now XML making a very welcome appearance in the Web world, and its maths application, MathML. Can we hope for a happy co-existence between MathML and \TeX ?
- 2 Unicode, and its superset standard ISO 10646-1. 16-bit encoding is a reality, and is alive and well in the \TeX world, with the extension of \TeX called Omega, by John Plaice and Yannis Haralambous. Among many other things, this will allow Indian \TeX users to typeset scripts such as Malayalam to a high standard without recourse to pre-processors or any limitations on transcriptions.
 - 3 Font control. Perhaps we are not in an ideal situation with regard to font encoding and usage in normal \TeX , but we are getting there. While in the \LaTeX world T1 encoding is widely accepted, we can hope for similar agreement on 8-bit maths font encoding.
 - 4 Pictures. Has the \TeX user ever had such a rich variety of ways to draw high-quality pictures? Choose from MetaPost, XYpic, or PSTricks for general purpose drawing, Musix \TeX for music, PPCH \TeX for chemistry, FeynMF for Feynman diagrams, or dozens of others.
 - 5 Macro packages. With the decline of most of the original plain-based macro packages put together in the mid-late 1980s, \LaTeX , with its huge collection of add-on packages, now has a slightly unhealthy domination of the \TeX -using world. Perhaps 1998 will be the time for some new competitors, such as CON \TeX t, to start a serious challenge?
 - 6 \TeX extensions. For a good many \TeX users, all that is really needed is the Plain format, and a few (just a few) tweaks to \TeX to solve some of those nagging programming issues. For some other users, what is needed is a complete re-implementation of \TeX in Java, to provide the basis for serious experimentation with modularisation and spreading resources across the Internet. The first group can rejoice, as the second version of $\epsilon\text{-TeX}$ is almost released, offering a variety of new features, without compromising \TeX 's integrity in any way. The second group can also start to uncross their fingers, as the NTS group will start work on a rewrite of \TeX in Java during 1998.
 - 7 \TeX implementations. Does any \TeX user still need to compromise on their implementation? With the standard Unix web2c implementation now available for DOS, Windows 32, Amiga, and any Unix platform, healthy competition in Windows 32 from Mik \TeX , ably maintained shareware Macintosh systems (Oz \TeX and CMac \TeX), it almost seems an embarrassment of riches to mention the commercial Y&Y \TeX , Textures, and Scientific Workplace, each offering very significant advantages.
 - 8 The relationship with SGML. We are reaching an understanding of how to use \TeX as a backend formatter for SGML, and also how to use it as a front-end, by converting \TeX mss to SGML. Not an easy area to work in, perhaps, but \TeX can hold its head high, and say that it is not afraid to work with SGML.
 - 9 DSSSL, the ISO 10179 standard for a 'Document Style Semantics and Specification Language', is now not only a reality, but in daily use, thanks to James Clark's free implementation, `jade`. By separating style specification from paragraph makeup, this keeps a firm place for \TeX in the universe, since DSSSL implementations (or the subsets, such as XSL, which may be implemented for the Web) need a batch-oriented formatting engine to assist them.
 - 9.8 User groups. Not a whole reason, because we do not yet have user groups to help \TeX users all over the world. But by adding the Indian \TeX Users Group, we are making a significant contribution to filling that missing 20%.

But a word of warning to conclude — let us not take \TeX into a ghetto. Let it work *alongside* other tools and technologies, not against them.

2 The 1997 AGM meeting

Those members who had the opportunity, or good sense, to come to the UKTUG annual general meeting had a rare treat of listening to five very different, but all entertaining and interesting, speakers:

1. Peter Flynn talked about what it *means* to archive documents, a very salutary reminder that there is more to preserving your work than occasionally copying `*.tex` on to a floppy disk ...
2. Hans Hagen described his work in developing the hugely-powerful \TeX macro package Con \TeX t, and some of the ways he uses it to produce interactive PDF documents, carefully designed for screen use;
3. Timothy Murphy meditated on \TeX and friends redone in Java, and offered a compelling vision of a network-enabled, distributed, \TeX world;
4. Han The Thanh gave an introduction to pdf \TeX , the \TeX variant which produces PDF instead of dvi;

5. Philip Taylor described the mouth-watering programmer's enhancements to be provided in ϵ - $\text{T}_{\text{E}}\text{X}$, and the plans for NTS.

Baskerville plans to bring you a flavour of all these presentations. In this issue we print articles by Phil Taylor and Han The Thanh; Peter Flynn and Tim Murphy should follow in the next *Baskerville*, and we hope to devote an entire issue to $\text{ConT}_{\text{E}}\text{Xt}$.

3 $\text{T}_{\text{E}}\text{X}$ Live 3

We are preparing to make a new $\text{T}_{\text{E}}\text{X}$ Live CD-ROM, with an anticipated release date of mid March 1998. It will be distributed to all UKTUG members. The changes and enhancements since last year include:

- Updated macro, font and documentation packages
- December 97 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$
- Improved install scripts
- Unix, Win32, DOS (DJGPP) and Amiga binaries based on web2c 7.2
- WIN95/NT system runnable directly from the CD-ROM
- Omega, ϵ - TeX and pdf TeX for Unix and Win32
- Latest C MacTeX , Oz TeX , and Mik TeX distributions
- Addition of the extra utilities lacheck, dvidvi, dtl, psutils, tlutils, dviselect, dviconcat and TeX4ht for all the Unix and Win32 binary sets
- The full $\text{ConT}_{\text{E}}\text{Xt}$ macro package
- Joliet CD-ROM support, so that win32 users see long/mixed-case filenames

We hope that these improvements make an attractive package for members, and take another step towards the painless free $\text{T}_{\text{E}}\text{X}$ installations many of us dream of.

II Chairman's Report to the AGM of UKTUG for the period October 1996 to November 1997

Robin Fairbairns

I have now served my two year 'stint' as chair of UKTUG and have not put my name forward for re-election: I shall explain my reasons later.

I would not claim that the last 13 months have been good ones for UKTUG — though I would maintain that we've done *some* things right. In my view, one principal reason for the shortcomings of the group's performance has been (again) the difficulty the committee has in 'covering the ground' of the tasks that need to be done. The committee has consisted of 6 people (plus me, as chair); this is half-strength. The effect is that if someone's energy flags on a particular task, there's little chance that there will be anyone else to chivvy them on to greater efforts. I urge everyone seriously to consider allowing their name to go forward for membership of the committee — let's give the new chair of UKTUG a good basis for his future work.

Principal among the things we *did* do right is Sebastian Rahtz's sterling work that has given us the T_EX Live CD (of which we sent every member a copy). T_EX Live was originally a UKTUG project, though the current (second) edition was supported by a number of other groups too; I believe it has had significant impact on the utility of T_EX on the increasingly many Unix-based machines throughout the world, many of which are in people's own homes. It has certainly been the largest impetus for people to join the group that I've experienced.

We agreed to support the TUG bursary fund for 1997, but we were never asked to submit monies. We have therefore decided to offer bursary money both for EuroT_EX'98 (in St. Malo — you will have seen the flier in a *Baskerville* mailing) and for TUG'98 (in Toruń, Poland). These bursaries are open to anyone who is sufficiently impoverished: as a further innovation, we have proposed a further bursary open to members of UKTUG only, for attendance either at EuroT_EX'98 or at TUG'98. In the traditional way, these bursary monies (though all are to be spent within Europe this year) are expressed in dollars: the UKTUG-member-only one is of \$500.00, and our contributions to the meeting bursaries themselves are of \$1,000.00 each.

Rather less positive has been our performance with *Baskerville*. You have seen only three editions since last we met (since there was already a backlog at that time, you should in principle have seen seven); of course, one of those three was a 'double' issue, but

To a significant extent, we are in your hands in this matter: your committee can't write *everything* that is to appear in *Baskerville*, yet the only other member who has submitted anything has been Andy Lack. What's more, despite a plea from me for 'frequent' questions to answer, only one member has submitted something for me to add to the FAQ list. We have formed a schedule for future issues of *Baskerville*, but the problem of filling them will recur.

I can now report that the incoming committee¹ has concluded that it is (in essence) impossible to retrieve the backlog of issues of *Baskerville*, and that any pretence of completing its volume 7 should be abandoned. We shall be sending a copy of the NTG's excellent 4AllT_EX CD to all 1996–97 members², and after that has been sent there will be no further mailings to our 1996–97 mailing list.

We have held no meetings (other than the present one). This was a conscious decision by your committee: as I explained at our last AGM, attendance at our meetings has been very poor of late, and tutorial meetings (which are what people actually ask for) are more poorly attended still, and regularly make a thumping loss. There is an agenda item for you to propose topics for meetings, but you can submit proposals at any time to the Secretary (see back page for contact details).

You will remember that last year I mentioned the committee's ambition to change UKTUG's financial year. The reason (as I explained last year) was to enable us to hold an annual meeting during the late part of the summer, combining a significant (two-day) technical meeting with a general meeting. Unless we change the financial year we cannot hope to conduct a sensible AGM any earlier in the year than October. Unfortunately, such a change would be a change to the constitution, and constitutional motions require infeasibly large attendance at meetings to get passed.

¹Of which I am a member: I concur with the decision

²Those of you that the group can send email to will already have heard of this decision

reprinted from Baskerville

Volume 8, Number 1

We have therefore decided not to present a motion on this matter, and the only motions your committee has submitted are those we are required by the constitution to submit.

III The pdfTeX user manual

Han The Thanh
Faculty of Informatics,
Masaryk University,
Brno,
Czech Republic
thanh@informatics.muni.cz

1 Introduction

The pdfTeX project started in 1996 at the Faculty of Informatics, Masaryk University, Brno, Czech Republic. It forms a primary part of the MSc and PhD research of Han The Thanh, under the supervision of Jiří Zlatuška and Petr Sojka. The main purpose of the project was to create an extension of TeX that could create PDF directly from TeX source files and improve/enhance the result of TeX typesetting with the help of PDF. pdfTeX contains TeX as a subset. When PDF output is not set, it produces normal DVI output; when PDF output is selected, pdfTeX produces PDF output that looks identical to the DVI output. The next stage of the project, apart from fixing any errors in the program, is to investigate alternative justification algorithms, possibly making use of multiple master fonts.

pdfTeX is based on the original TeX sources and web2c, and has been successfully compiled on Unix, Amiga, Win32 and DOS systems. It is still under beta development and all features are liable to change.

This manual was compiled by Sebastian Rahtz from notes and examples by Han The Thanh. Many thanks are due to members of the pdfTeX mailing list (most notably Hans Hagen), whose questions and answers have contributed much to this manual.

2 Implementation details

The implementation of pdfTeX consists of two parts: the changes to TeX, and the addition of ‘driver’ features. The TeX-specific part is written as a change file to the original source of TeX and contains:

- a part that generates PDF code visually equivalent to DVI commands;
- virtual font processing;
- implementation of new primitives for PDF control.

The ‘driver’ part is written as several source files in C, and implements

- font mapping,
- font inclusion and partial downloading,
- font re-encoding,
- PNG picture insertion (using the public domain libpng code),
- text compression (using the public domain zlib code).

2.1 Compilation

pdfTeX is supplied as a set of additions to the standard Unix Web2c setup, and is a standard part of Web2c 7.2 (from January 1998); compilation should present no problems on most Unix systems. Porting it to other setups will require a little work, because of the requirement to combine the normal TeX Web output, and the parts written in C.

For Win32 systems (Windows 95, Windows NT) there are two packages that contain pdfTeX, both in CTAN: systems/win32. Web2c for Win32 is maintained by Fabrice Popineau (mailto:popineau@ese-metz.fr), and MikTeX by Christian Schenk (mailto:cschenk@berlin.snafu.de).

A binary version of pdfTeX for Amiga is made available (CTAN:systems/pdftex/bin/Amiga) by Andreas Scherer (mailto:Scherer@Physik.RWTH-Aachen.De).

To make the situation more complicated, a Win32 binary version of pdfTeX compiled with Cygnus tools will also be made available in CTAN:systems/pdftex/bin/Win32. This version will be compiled in the same way as Unix systems.

2.2 Search paths

When running pdfT_EX, some extra search paths need to be set up beyond those normally requested by T_EX itself; in web2C these are:

VFFONTS the path where pdfT_EX looks for virtual fonts.

TIFONTS the path where pdfT_EX looks for Type1 and TrueType fonts.

TEXPSHEADERS the path where pdfT_EX looks for the font mapping file (`pdfTex.map`), PNG pictures and encoding files (`*.enc`).

3 Fonts

pdfT_EX can only work with Type 1 and TrueType fonts at present, and a source must be available for all fonts used in the document, except for the 14 base fonts supplied by Acrobat Reader (Times, Helvetica, Courier, Symbol and Dingbats). It is *not* possible to use METAFONT-generated fonts in pdfT_EX—apart from the technical reasons, the resulting Type 3 fonts render very poorly in Acrobat Reader. Given the free availability of Type 1 versions of all the Computer Modern fonts, and the ability to use standard PostScript fonts without further ado, most existing T_EX users should be able to experiment with pdfT_EX.

pdfT_EX reads a map file called `pdfTex.map`. Reencoding and partial downloading for each font is specified in this file. Every font must be listed in this file, each on a separate line. The syntax of each line is similar to dvips' map files (but may be changed later). Each line contains the following fields (some of them are optional):

```
texname  basename  fontflags  fontfile  encoding
```

texname: the name of the TFM file

basename: the base font name (PostScript font name)

fontflags: an integer number specifying the most important characteristics of the font. This number is significant only in the case that the font file is *not* included, and Acrobat Reader is asked to simulate missing font using its multiple master defaults. It is a PDF font descriptor (see PDF manual, section 7.9.2) which is a 32-bit integer containing a collection of Boolean attributes, with bit 1 being the least significant. Attributes are true if the corresponding bit is set to 1 in the integer. The meanings of the bits is as follows:

- 1 Fixed-width font
- 2 Serif font
- 3 Symbolic font
- 4 Script font
- 5 (reserved)
- 6 Uses the Adobe Standard Roman Character Set
- 7 Italic
- 8–16 (reserved)
- 17 All-cap font
- 18 Small-cap font
- 19 Force bold at small text sizes
- 20–32 (reserved)

Bit 6 indicates that the font's character set is the Adobe Standard Roman Character Set, or a subset of that, and that it uses the standard names for those characters. About bit 19, the PDF specification says

... used to determine whether or not bold characters are drawn with extra pixels even at very small text sizes. Typically, when characters are drawn at small sizes on very low resolution devices such as display screens, features of bold characters may appear only one pixel wide. Because this is the minimum feature width on a pixel-based device, ordinary non-bold characters also appear with one-pixel wide features, and cannot be distinguished from bold characters. If bit 19 is set, features of bold characters may be thickened at small text sizes.

It should be stressed that the font flags provided for many fonts in the currently distributed `pdfTex.map` are not correctly derived.

fontfile: the name of the font source file. This must be a Type 1 or TrueType font file. If it is preceded by a < then the font file will be partly downloaded; if it preceded by a double << then the font file will be included entirely.

Otherwise the font is not included, and only the font parameters are extracted. Note that the font file *must* always be available at runtime, even if it not downloaded.

encoding: the encoding vector used for the font. It may be preceded by a <, but the effect is the same. The format is identical to that used by dvips.

Here are some sample lines from pdftex.map:

Include font entirely without reencoding —

```
pgsr8r GillSans 34 <pgsr8a.pfb
```

Include font partly without reencoding —

```
pgsr8r GillSans 34 <pgsr8a.pfb
```

Do not include font, or reencode it, but extract parameters from font file —

```
pgsr8r GillSans 34 pgsr8a.pfb
```

Include font entirely and reencode —

```
pgsr8r GillSans 34 <pgsr8a.pfb 8r.enc
```

Partially include font and reencode —

```
pgsr8r GillSans 34 <pgsr8a.pfb 8r.enc
```

Do not include font but extract parameters from font file and reencode —

```
pgsr8r GillSans 34 pgsr8a.pfb 8r.enc
```

A TrueType font can be used in the same way as a Type 1 font —

```
Cxr10 Cxr10 34 <Cxr10.ttf Cxtext.enc
```

A base font can also be reencoded; the name of the font file is simply ignored —

```
phvr8r Helvetica 32 <Helvetica.pfb 8r.enc
```

4 New primitives

Here is a short description of new primitives added by pdfTeX:

`\pdfoutput=n`

Integer parameter specifying whether the output format should be DVI or PDF. Positive value means PDF output, otherwise DVI output. This parameter cannot be specified *after* shipping out the first page.

`\pdfcompresslevel=n`

Integer parameter specifying the level of text compression via `zlib`. Zero means no compression, 1 means fastest, 9 means best, 2..8 means something in between.

`\pdfpagewidth=dimen`

`\pdfpageheight=dimen`

Dimension parameters specifying the page width and page height of PDF output. If not specified then they are calculated as `\hsize` or `\vsize + 2 × (1in + \hoffset or \voffset)`, when pdfTeX writes the first page.

`\pdfliteral{pdf text}`

Like `\special` in normal TeX, this can be used to insert raw PDF code into the output. This allows support of color, and text transformation.

`\pdfinfo author{author} title{title} subject{subject} keywords{keywords}`

Specify document information. All keys are optional; if this information is provided, it can be seen in Acrobat Reader with the menu option Document Information ~> General

`\pdfcatalog pagemode{pagemode} uri{uri}`

Document display information. Both keys are optional. *URI* provides the base URL of the document, and *pagemode* determines how Acrobat displays the document on startup. The possibilities are:

<code>/UseNone</code>	Open document with neither outline nor thumbnails visible.
<code>/UseOutlines</code>	Open document with outline visible.
<code>/UseThumbs</code>	Open document with thumbnails visible.
<code>/FullScreen</code>	Open document in full-screen mode. In full-screen mode, there is no menu bar, window controls, nor any other window present.

The default is `/UseNone`.

`\pdfimage height height width width filename`

Insert a bitmap image in PNG format, optionally changing width, height or both. Dimensions which are not specified will be treated as zero. If both height and width are zero then the box takes the natural size of the image. If one of them (width or height) is zero and the second is not, then the first one (the zero one) will be set to a value proportional to the second one so as to make the box have the same proportion of width and height as the image natural size. If both width and height are positive then the image will be scaled to fit these dimensions.

`\pdfform num`

Write out the \TeX box *num* as a Form object to the PDF file.

`\pdflastform`

Returns the object number of the last Form written to the PDF file

`\pdfrefform \name`

Put in a reference to the PDF Form called `\name`.

These macros support a feature called “object reuse” in pdf \TeX . The idea is to create a Form object in PDF. The content of the Form object corresponds to the content of a \TeX box, which can also contain pictures and references to other Form objects as well. After that the Form can be used by simply referring to its object number. This feature can be useful for large document with a lot of similiar elements, as it can reduce the duplication of identical objects.

`\pdfannottext width width height height depth depth attr{attr} {text}`

Attach a text annotation at the current point in the text. The attributes can be used to specify the default appearance of the annotation (e.g., `/Open true` or `/Open false`), as well as many other features (see Table 6.8 of the PDF manual).

`\pdfdest num num name {name} appearance`

Establish a destination for links and bookmark outlines; the link must be identified by either a number or a symbolic name, and the way Acrobat is to display the page must be specified; *appearance* must be one of

<code>fit</code>	fit whole page in window
<code>fith</code>	fit whole width of page
<code>fitv</code>	fit whole height of page
<code>fitb</code>	fit whole ‘Bounding Box’ page
<code>fitbh</code>	fit whole width of ‘Bounding Box’ of page
<code>fitbv</code>	fit whole height of ‘Bounding Box’ of page

`\pdfannotlink height height depth depth attr{attr} action`

Start a hypertext link; if the optional dimensions are not specified, they will be calculated from the box containing the link. The *attributes* (explained in great detail in section 6.6 of the PDF manual) determine the appearance of the link. Typically, this is used to specify the color and thickness of any border around the link. Thus `/C [0.9 0 0]` `/Border [0 0 2]` specifies a color (in RGB) of dark red, and a border thickness of 2 points.

The *action* can do many things; some possibilities are

goto num *n*
 goto name {*refname*} Jump to a point established as *name* with `\pdfdest`
 goto file {*filename*} Open a local file; this can be used *with* a *name* or *num* specification, to point to a specific location on the file. Thus `goto file{foo.pdf} name{intro}`
 thread num {*n*}
 thread name {*refname*} Jump to thread identified by *n* or *refname*
 user {*spec*} Perform user-specified action. Section 6.9 of the PDF manual explains the possibilities. A typical use of this is to specify a URL, e.g. `/S /URI /URI (http://www.tug.org/)`.

`\pdfendlink`

Ends link; all text between `\pdfannotlink` and `\pdfendlink` will be treated as part of this link. pdfTeX may break the result across lines (or pages), in which case it will make several links with the same content.

`\pdfoutline action count count {text}`

Create a outline (or bookmark) entry. The first parameter specifies the action to be taken, and is the same as that allowed for `\pdfannotlink` (see above, though note that the `Page` key does not work properly at present). The *count* specifies the number of direct subentries under this entry, 0 if this entry has no subentries (in this case it may be omitted). If the number is negative, then all subentries will be closed and the absolute value of this number specifies the number of subentries. The *text* is what will be shown in the outline window (note that this is limited to characters in the PDFEncoding vector).

`\pdfthread num num name {name}`

Start an article thread; the corresponding `\pdfendthread` must be in the box in the same depth as the box containing `\pdfthread`. All boxes in this depth level will be treated as part of this thread. An identifier (*n* or *refname*) must be specified; threads with same identifier will be joined together.

`\pdfendthread`

Finish the current thread.

`\pdfthreadoffset=dimen`

`\pdfthreadvoffset=dimen`

Specify thread margins.

One way to learn more about how to use these primitives is to have a look at the file `example.tex` in the pdfTeX distribution.

5 Graphics and color

Probably the biggest single usage problem with pdfTeX at the present time is the inclusion of graphics. The program only directly supports graphic inclusion in one bitmap format, PNG (Portable Network Graphics).

Two commonly-used techniques are not available — the inclusion of Encapsulated PostScript figures, and the inclusion of raw PostScript commands (the technique utilized by the `pstricks` package). Although PDF is a direct descendant of PostScript, it lacks any programming language commands, and cannot deal with arbitrary PostScript. There are two ways to proceed with existing EPS files: firstly, convert them to PNG (using programs like Image Magick, Image Alchemy, or Ghostscript); or secondly, try converting them to simple PDF. If the picture has no special fonts, the chances are quite good that Ghostscript's pdf writer will produce a file containing a single PDF object, which can be included using `\pdfliteral` commands (this is managed by the standard L^AT_EX graphics package).

Other alternatives for graphics in pdfTeX are:

1. L^AT_EX picture mode: since this is implemented simply in terms of font characters, it works in exactly the same way as usual;
2. Xy-pic: If the PostScript backend is not requested, Xy-pic uses its own Type 1 fonts, and needs no special attention;
3. tpic: The `'tpic'` `\special` commands (used in some macro packages) can be redefined to produce literal PDF, using macros by Hans Hagen;

4. MetaPost: although the output of MetaPost is PostScript, it is in a highly simplified form, and a MetaPost to PDF conversion (written by Hans Hagen and Tanmoy Bhattacharya) is implemented as a set of macros which read MetaPost output and support all of its features;
5. It is possible to insert a “pure” PDF file (PDF that has only one page without fonts, bitmaps or other resources) using a macro package that reads the external PDF file line by line.

The two latter macro packages are part of CONTEX (supp-pdf.tex and supp-mis.tex), but also work with L^AT_EX and are distributed separately.

For new work, the MetaPost route is highly recommended. For the future, Adobe have announced that they will define a specification for ‘encapsulated PDF’, and this should solve some of the present difficulties.

6 Macro packages supporting pdf_TE_X

- For L^AT_EX users, Sebastian Rahtz’ hyperref package has substantial support for pdf_TE_X, and provides access to most of its features. In the simplest case, the user merely needs to load hyperref with a ‘pdftex’ option, and all cross-references will be converted to PDF hypertext links. PDF output is automatically selected, text compression turned on, and the page size is set up correctly. Bookmarks are created to match the table of contents.
- The standard L^AT_EX graphics and color packages have pdftex options, which allow use of normal color, text rotation, and graphics inclusion commands. Only PNG and MetaPost files can be included.
- The ConTEX macro package by Hans Hagen (mailto:pragma@pi.net) has very full support for pdf_TE_X in its generalized hypertext features.
- Hypertexted PDF from texinfo documents can be created with pdftexinfo.tex, which is a slight modification of the standard texinfo macros. This is part of the pdf_TE_X distribution.
- A similiar modification of the webmac, called pdfwebmac.tex, allows production of hypertexted PDF versions of program written in WEB. This is part of the pdf_TE_X distribution.

Some nice samples of pdf_TE_X output can be found on the TUG Web server, at <http://www.tug.org/applications/pdftex/>.

7 ϵ - \TeX V2: a peek into the future

Philip Taylor
Royal Holloway and Bedford New College,
University of London
E-mail: P.Taylor@Vms.Rhbnc.Ac.Uk

Summary

ϵ - \TeX V1 was released towards the end of 1996, and it was intended at the time that ϵ - \TeX V2 should be released approximately one year later. For various reasons the release date has slipped a little, but V2 is in alpha-test and we confidently expect to release it in the near future: indeed, it may well have been released by the time that this article appears in print. In this article the new features of V2 are reviewed, and we conclude by peeking a little further into the future to see what will happen with NTS.

8 Introduction

When the NTS project was first established in 1992, we hoped that work would commence fairly quickly. Sadly it became all too obvious that a project of that magnitude would require one or more full-time workers, and none of the (volunteer) team had that sort of time to spare. The team decided that, until funding could be found, they would work on the less-demanding but still worthwhile task of extending \TeX in its current (Pascal-WEB) form. The first fruits of that work were revealed in late 1996, when ϵ - \TeX V1 was released. This version added approximately 30 new primitives to \TeX and a small adjunct macro library was also produced which both extended the plain \TeX format to accommodate the new primitives and also added new functionality such as natural language handling and \TeX module libraries.

Once ϵ - \TeX V1 had been released, the group were free to concentrate on the next version. Originally intended to ship a year later than the first, this version has slipped a little as pressure of other commitments has forced members of the team to invest less time in the project than they would have wished. However, V2 is now in alpha test, and we confidently expect to be able to make a general release in the near future: release may well have taken place by the time this article appears in print. The majority of the remainder of this article describes the features that we are fairly confident will be present in that release.

9 Ideas which are almost certain to appear in ϵ - \TeX V2

Although we do not wish to give an absolute commitment at this stage, particularly as many of the proposals are the process of being tested at the time of writing, we do believe that the ideas contained in the following section are very likely to appear in ϵ - \TeX V2. A subsequent section discusses ideas which may appear in future releases.

9.1 *Increasing \TeX 's registers*

\TeX has 256 of the most commonly used registers: counts, dimens, skips, boxes, toks, etc., and whilst these are enough for normal applications, advanced formatting systems really require more. In ϵ - \TeX V2, we intend to provide 32768 of each of these, which we hope will be sufficient for the most demanding packages. Insertion classes will still be restricted to 256 or fewer, and $\backslash\text{box } 255$ will retain its special significance. The 'etex' format will allow both local and global allocation of these registers ('plain' allows only global), and for efficiency reasons a user will be able to elect whether to allocate a register from the dense (0..255) pool or from the sparse (256..32767). To allow the allocation mechanism to overflow from dense to sparse without risking a conflict with the allocation of insertion classes, the format allows a user or package to pre-reserve a number of insertion classes. Facilities for block-allocating a contiguous set of registers will be provided.

9.2 *Improved natural language handling*

\TeX overloads the $\backslash\text{lccode}$ concept, using it both for 'real' lower-casing operations and also for purposes of hyphenation. In ϵ - \TeX V2 these operations are unbundled, and the codes used for hyphenation can be staticised as the patterns are read in (the current set of lccodes is used). Thereafter, whenever a particular language is used, the corresponding set of hyphenation codes is loaded.

9.3 Arithmetic expressions

Although \TeX can perform simple arithmetic (addition, multiplication and division), these operations are ‘assignments’ and therefore cannot be used in expansion-only and certain other contexts. $\epsilon\text{-}\TeX$ V2 provides a set of arithmetic primitives which evaluate an expression in such a way that the value of the expression can be accessed in expansion-only contexts, as well as being usable (for example) when \TeX is looking for a $\langle number \rangle$, $\langle dimen \rangle$, etc. As \TeX intentionally uses only integer arithmetic wherever the results of a computation are accessible to the user, floating-point arithmetic has *not* been provided. There are four new primitives, `\numexpr`, `\dimexpr`, `\glueexpr` and `\muexpr`, each of which requires its operands to be of appropriate type (or coercible to that type). Parentheses may be used to indicate precedence wherever this will clarify or disambiguate an expression. The normal arithmetic operators ‘+’, ‘-’, ‘*’ and ‘/’ are allowed within an expression.

9.4 Discards are no longer discarded!

When \TeX performs page-breaking, so-called ‘discardable items’ which follow the chosen breakpoint are discarded; whilst this is perfectly reasonable if the page break is actually taken, recursive techniques aimed at optimising the appearance of multiple pages require the ability to ‘undo’ a pagebreak in order to try the effect of breaking elsewhere. The discarded items are therefore required in order to re-create the vertical list which \TeX is trying to break. In $\epsilon\text{-}\TeX$ V2, we allow access to these ‘discarded items’ via a new primitive `\pagediscards`. The discards which occur during `\vsplitting` are also accessible via an analogous primitive `\splitdiscards`. Both of these primitives return a vertical list, similar to that obtained by `\unvboxing` a box register.

9.5 Read-write access to `\parshape`

Although \TeX allows the user to create arbitrarily complicated paragraph shapes through the use of the `\parshape` primitive, it provides no way for the user to find out which `\parshape` is currently active (although it does allow the user to ascertain the number of lines of the current `\parshape` specification). In $\epsilon\text{-}\TeX$ V2, we allow full read access to all the elements of the current `\parshape`.

9.6 Interrogating the current conditional context

In $\epsilon\text{-}\TeX$ V1 we allowed users to make environmental enquiries concerning the current group, both as to its depth of nesting and to its type. In $\epsilon\text{-}\TeX$ V2, we generalise this concept and allow analogous access to the current conditional context through the use of `\currentiflevel`, `\currentiftype`, `\currentifbranch` and `\showifs`.

9.7 Access to information concerning font-character combinations

Although it is possible to gain some information about a particular character in a given font by typesetting that character in a box and then measuring the dimensions of the box, not all the dimensions of the character can be reliably obtained in this way, and there is no way to ensure that the character actually exists in the font before attempting to typeset and measure it. In $\epsilon\text{-}\TeX$ V2 we allow the user both to check whether a particular character exists in a given font, using `\iffontchar`, and (if it does exist) to measure the four fundamental dimensions of that font/character combination using `\fontcharwd`, `\fontcharht`, `\fontchardp`, and `\fontcharic` (representing width, height, depth and italic correction respectively). Furthermore, we ensure that users are alerted to the existence of missing characters in a font by causing lost characters to be logged to the console as well as to the log file if `\tracinglostchars` is set to a value greater than 1.

9.8 Better debugging aids

In order to assist in diagnosing mis-matched or runaway group problems, $\epsilon\text{-}\TeX$ V2 allows the user to opt to be warned whenever a file is left in a group or conditional other than that at which it was entered. This may be accomplished by setting `\tracingnesting` to a value greater than zero.

9.9 Subtle change to the semantics of `\protected`

$\epsilon\text{-}\TeX$ V1 introduced a new prefix, `\protected`, which inhibited the expansion of the ‘protected’ macro in contexts in which expansion was unlikely to be required. Further research into this area suggested that at least one such case had been missed, and ‘protected’ macros are now inhibited from expansion when \TeX is scanning ahead while processing alignments.

9.10 Optimisations

To improve the overall efficiency of $\epsilon\text{-}\TeX$ internal modifications have been made to reduce the resources required when there are a number of `\aftergroups` active for a single group, and to eliminate the stack space wasted in setting a register to the same value as it currently holds.

9.11 Access to the components of a glue quantity

Whilst it is possible to gain access to the various components of a glue value by clever macro programming, the code required is sufficiently arcane to suggest that a better method is much to be preferred. Accordingly we are considering a set of primitives `\gluestretch`, `\gluestretchorder`, `\glueshrink` and `\glueshrinkorder` which will give much-simplified access to these quantities. As a part of the same process we are looking at two conversion primitives, `\mutoglu` and `\gluetomu`.

9.12 Improved typographic quality

Whilst the majority of the work in ϵ -T_EX is aimed at providing the ϵ -T_EX programmer with more powerful tools, we are aware that the real purpose of T_EX is to generate typeset output of the highest quality. During a meeting in Brno with Prof. Knuth on the occasion of his honorary doctorate, he suggested that we might like to consider improving the typographic quality of the last line of a paragraph. According to Don, traditional (hot-lead) typesetters would set the last line to the same tightness or looseness as the immediately preceding line, and he thought that ϵ -T_EX should be capable of doing likewise. We are looking into providing this but in a parameterised manner, so that *all* possibilities between T_EX's current behaviour and that suggested by Don can be achieved. We think that this might be controlled by a parameter called `\lastlinefit`.

9.13 Improved typographic quality, cont.

In the same vein, we are looking into ways of allowing better parameterisation of the page-breaking process by having not just one penalty for (say) a club- or widow line but a whole array of such penalties which can reflect the undesirability of leaving one, two, three to n lines at the top or bottom of a page. Other related penalties are also candidates for this process.

10 Ideas still under discussion

The following ideas are all under discussion but are very unlikely to find their way into ϵ -T_EX V2: some may be deferred to ϵ -T_EX V3, and some may never appear at all. Although the group have some idea into which category each of these ideas may fall, it is probably not helpful to go into too much detail here, and so they are all lumped together as 'under discussion'.

10.1 Can T_EX find this font?

In 'the good old days', a T_EX program could count on finding all 76 of the standard T_EX fonts no matter where it was run in the world. These days, with many documents being set in exotic fonts from a myriad of sources, it is no longer certain that, just because site A has font F, site B will have the same font. We are therefore considering providing an `\iffont` primitive which will allow ϵ -T_EX to ascertain at run-time whether a particular font exists on the system on which the document is being processed. It is not certain at this stage whether this would be a simple 'does this font exist?' test, or a more complex 'does it exist and is the TFM file for it valid?'. `\tryfont` has also been suggested as an alternative approach.

10.2 Maths alignments

Peter Breitenlohner, the implementer of ϵ -T_EX, probably typesets more mathematics than the rest of the group put together, and he believes that there is a case for a maths alignment primitive. He has not yet finished his research on this topic, and all that can be said at this stage is that we are considering implementing some form of `\mathalign`.

10.3 Typesetting on a grid

Whilst T_EX is *excellent* at typesetting in designs where variable quantities of white space can be allowed to occur, trying to coerce it to set on a regular grid (something at which packages such as Quark Xpress excel) is *far* more difficult. The various macro-based solutions which have been tried do not seem to address the underlying problems, and we are looking at providing an entirely new paradigm within ϵ -T_EX whereby material being typeset can be caused to 'lock on' to a grid at some point in the page-building process. Although at first sight it might be thought that it is the reference points of the lines making up the page which need to lock to this grid, we are fairly certain that this is not always the case, and we are therefore looking at ways of associating one or more 'handles' with a particular box. In the degenerate case there will be one handle which is coincident with the reference point of the line, but in more complex cases there may be two, three or ever more handles, each of which will lock on to one line of the grid. Even so, there are also situations in grid-based designs where the grid-lock constraints just have to be violated, and one topic still unresolved is whether it is then better simply to allow the box to float free, or whether it is better to constrain it in some way, perhaps by associating with the handle(s) a degree of flexibility which is in some way analogous to T_EX's current use of the 'glue' concept.

Subsection More on improved typographic quality Another point made by Don during his stay in Brno is that there are situations in which \TeX 's (vertical) positioning of elements of mathematical formulæ is less than ideal. He points out that even in typesetting the \TeX book he had to make use of kludges such as `\sub` `\strut` in order to achieve the best visual effect. We are investigating ways in which the effect (and related effects) could be achieved by better parameterisation of the mathematical typesetting process.

11 NTS

In the introduction to this paper it was mentioned that the NTS project proper had been put 'on ice' until the group had sufficient funds to allow a programmer to be employed full-time to work on the project. It is with great pleasure that I can now report that, as a result of the generosity of DANTE e.V. the group has DM 30 000 which can be used for this purpose. On the recommendation of Jiří Zlatuška, we have made an offer to Karel Skoupy of the Czech Republic, which he has accepted, and Karel will be starting work on NTS during late February 1998. It has been agreed that the language of implementation will be Sun's Java, and Karel's first task (apart from becoming a Java expert . . .) will be to draw up a specification for NTS. Provided that the group agree with his design, he will then start work on implementing NTS, and we hope to be able to review his work after a further six months. Within one year of commencement we hope to have a working implementation of NTS, not simply a port of \TeX to Java but a total re-design intended to emphasise the deep structure of \TeX whilst avoiding the design features which make the present system rather difficult to extend or change.

The group are still determined that NTS will be 100% \TeX -compatible, and are confident that it will remain so for at least the first five years of its life. We are less certain whether divergence should then be permitted, in order to add new functionality which is in some way incompatible with \TeX . If we do decide that compatibility must be sacrificed, we will give considerable notice of that decision, and users who must retain the ability to process legacy documents in a manner identical to \TeX will be advised to take an archive copy of NTS before compatibility is lost.

One exciting idea which the use of Java permits is the possibility to integrate access to `\` (CNAN?!) with NTS: it is by no means impossible that NTS might be able to fetch for itself any module which cannot be found on the local system and which is needed in order to process a document. If that becomes a reality, \TeX will have become truly integrated into today's (and tomorrow's) globally-networked world.

12 The inaugural meeting of TUG India

Sebastian Rahtz
Elsevier Science Ltd
Oxford
s.rahtz@elsevier.co.uk

13 First stirrings

Back in the summer when I first started corresponding with C. V. Radhakrishnan in India about \TeX and SGML-related matters, I little thought that I would be escaping the English winter for a week in Southern India at the start of 1998. But something seemed to crystallize in the minds of some Indian \TeX ies, and events moved fast in the subcontinent during the autumn. By November 16th, Radhakrishnan was able to announce to the world that the newest \TeX user group had been born:

The Indian \TeX Users Group has been informally launched today at the academic premises of Department of Mathematics of University of Kerala, Trivandrum. Prof. KSS. Nambooripad, a world-renowned mathematician and an ace \TeX programmer chaired the session. He was unanimously elected as the Chairman of the Indian \TeX Users Group. Following are the office-bearers of the TUGIndia.

Chairman: Prof. (Dr.) K. S. S. Nambooripad

Secretary: C. V. Radhakrishnan

Treasurer: Dr. R. Rajendra

Executive: Dr. A. R. Rajan (University of Kerala)
Dr. E. Krishnan (University College, Trivandrum)
Dr. V. N. Krishnachandran (Vikram Sarabhai Space Center)
Dr. R. K. Chettiar (Department of Education, Govt. of Kerala)
Mr. C. V. Rajagopal (University Observatory)
Mr. Deepak Tony Thomas, Oracle Corporation, Bangalore
Dr. P. Rameshkumar (MG University, Kottayam)
Dr. SRP. Nayar (Inter Univ. Center for Astronomy, Pune)

At the same time, they did me the great honour of inviting me to inaugurate the group, and I lost no time in accepting in principle. In the ordinary course of events it would have been beyond the finances of either TUG India or myself to pay for a trip there, but then a fairy godmother appeared, in the shape of the UK \TeX Users Group. The committee considered my tentative suggestion, and agreed that support of such a potentially important group would be a reasonable use of group funds. That just meant fixing a date, and finding a flight, and all was in train. By an amazing coincidence, another member of the UK TUG committee, Kaveh Bazargan, had already booked a holiday over Christmas and New Year at precisely the location in India chosen for the TUG India launch, so we were able to mount an even more impressive presence.

14 India, and the inauguration

The TUG India meeting took place in Trivandrum, the capital of Kerala state, which forms the southwest corner of India. It is a tropical area, which sometimes seems to be entirely covered in cocconut palms and banana trees, and is famous for its Communist state government, its almost 100% literacy rate, and a general air of some prosperity and a good distribution of wealth.

The principle mistake I made before setting off was to contract a vile cold, which rendered me almost speechless during my first few days, and a poor picture of health for the whole stay. However, after a long flight from London, a sweaty wait in Mumbai, and then a short flight to Trivandrum, it was little hardship to be taken off after lunch to the excellent beach resort of Varkala. Talking \TeX beneath the palms next to a sunny beach of the Arabian sea was a little disconcerting, but we managed

On Monday January 5th, we gathered in the University Observatory (a purely courtesy title these days) for the
reprinted from Baskerville *Volume 8, Number 1*

opening ceremony of TUG India, at which Kaveh and I were joined as speakers by Professor Nambooribad, the group's chairman, the University Vice Chancellor, and the local member of Parliament (showing a healthy interest in IT matters). Kaveh and I tried to present the T_EX world as place of dynamism, excitement and new possibilities for conventional and electronic publishing, and some at least of our audience seemed convinced.

Many of the delegates were from typesetting companies (some of them suppliers to my own employer, Elsevier), with the biggest contingent from Madras — Thomson's office seemed to have sent almost all their R&D team. But there was plenty of academic interest too, and of course a special concern with typesetting Indian scripts.

It was a pleasure to be able to hand over to Radhakrishnan a selection of T_EX-related books and journals, donated by Jonathan Fine, Malcolm Clark and myself, and to confirm the imminent despatch of back issues of TUGboat and MAPS to India. NTG had already sent a generous batch of 4AllT_EX CD-ROMs, of which each delegate was given a copy.

15 The first TUG India courses

On January 6th the serious work started, four days of tutorials in the morning on 'advanced' topics, and introductory L^AT_EX in the afternoons. I managed to avoid teaching the latter (I always find myself *apologizing* too much for L^AT_EX), but had fun in the mornings.

We started by dealing with a subject dear to my heart, and to that of some of the delegates — L^AT_EX to SGML translation. I expounded the Elsevier system, based on four stages of transformation:

1. L^AT_EX to dvi, using a very specialized class file, which redefines almost everything to put SGML markup into the dvi file;
2. dvi to ASCII (using Tobin's dtl programs);
3. ASCII to SGML against an intermediate DTD;
4. SGML to SGML for the final DTD (using a Perl library with directly interfaces with the NSGMLS parser).

It turned out that at least two others present had also thought of similar methodologies, which was reassuring.

From L^AT_EX to SGML, I moved on next day to DSSSL (Document Style Semantics and Specification Language) and its relationship with T_EX — perhaps not everyone present quite went along with me on that one. We were on safer ground discussing general aspects of electronic publishing using T_EX, and I was glad to be able to describe pdfT_EX in some detail, to publicize what I consider a much under-rated alternative to L^AT_EX2html (Eitan Gurari's T_EX4ht), as well as give a puff for my own L^AT_EX hyperref package.

On the third day, we moved onto pictures, and I attempted to make a (rather shaky) case for MetaPost. Colour was a subject where it was easier to find common ground, albeit by agreeing that color separation specification in T_EX was much too immature at present.

For the last day, I had decided that this was the moment where I would really make a first go at using Omega, and (somewhat to my surprise), I was able to write, compile and use a one-line Omega Transformation Process after some study of Omega examples. Since one of the Omega authors (Yannis Haralambous) is very actively working on the necessary OTPs, hyphenation and so on for typesetting Malayalam (the language spoken in Kerala), we can expect rapid deployment of Omega amongst those typesetting things like school textbooks.

16 ... and some sightseeing

After talking T_EX for 5 days, I was ready for some relaxation. We started with a shopping expedition, during which I bought some dresses for my daughters which are certain to lighten up wintry Oxford, and a selection of South Indian classical music. Then on the Saturday we drove across the state line into Tamil Nadu to visit the Padmanabhapuram palace of the Maharajah of Travancore, the princely state which occupied much of what is now Kerala until Independence. In the late 18th century, a replacement palace was constructed in Trivandrum, and Padmanabhapuram was left untouched. With elements from the 16th century, it is an incredible structure built almost entirely of teak, often intricately carved, and all ingeniously designed to keep the rooms cool with natural air-conditioning. Whether it was the ladies bathing tank, the audience chamber, or the hall where 2000 Brahmins could be entertained to dinner, the whole place was a marvel of design — and preservation by the State Archaeological Service! Perhaps the best moment was when we were granted special access to view the Maharajah's private meditation apartment whose plaster walls were covered in marvellous paintings, and where a pair of coconut-oil lamps had been burning non-stop for 200 years.

From the past to the present, as we drove to Cape Comorin, the southern-most tip of India, where you can see both sunrise and sunset across the sea from the same spot, and where three oceans meet. Here, in the late 19th century, Swami Vivekananda (a very influential religious reformer) swam out to a bare rock in the sea, meditated for five days,

and achieved a state of enlightenment to accord him the status of a saint. Now there is a modern memorial on the rock, and we joined hundreds of pilgrims in the boat ride to examine the spot. Thence back north, trying to visit a Jain temple set deep in a cave, but sadly the gates were locked, and some monkeys laughed from the rock.

On the Sunday, to Kerala's secret paradise, the long salt waterway that runs for 200 km parallel to the sea, sometimes as wide as a lake, at other times turning into quiet green tunnels with barely enough depth for the boat. A vista of endless coconut palms, half-hidden houses, and small fishing boats provided a very relaxing boat trip.

17 Conclusion

This was a worthwhile, if exhausting, trip, and I hope it gave a good start to TUG India. When I left, they already had 79 members signed up, just from word of mouth, so the group looks set to be active. It is hoped to cycle the meetings around the different parts of India, as well as publishing a newsletter, so the current bias towards the south should soon be corrected.

I must, of course, take this space to extend the heartfelt thanks of Kaveh and myself to C. V. Radhakrishnan and the many others who looked after us so magnificently during our stay in Kerala. They were very worthy ambassadors of a lovely part of India. I look forward to working with them, and hopefully to visiting India again soon.

TUGIndia can be contacted as follows:

C K Radhakrishnan
Secretary
TUGIndia
Kripa, TC 24/548, Sastha Gardens
Thycaud, Trivandrum 695014, India

Tel. +91 471 324341
Fax. +91 471 333186

Email: tugindia@mailexcite.com

18 An introduction to LyX

Andrew F. Lack
City University
sa346@city.ac.uk

19 What is LyX?

Am I a dinosaur living beyond my ‘best before’ date? Sometimes feels like it. To start with I use Unix. Every popular computing magazine you pick up today is filled with one word, and it isn’t Unix. Then I use L^AT_EX under Unix. People look at you as if you are from another planet. L^AT_EX? You mean you can’t *see* what it looks like as you type? But then I come across something like LyX and my faith in humanity, or that part of it which still uses Unix and L^AT_EX, is restored.

LyX is a full-featured “near” WYSIWYG editor for creating and editing L^AT_EX documents. Surprisingly, it’s not currently available from the CTAN archive, but by anonymous ftp from `ftp.via.ecp.fr/pub/lyx`. Binaries are available for some popular varieties of the Unix system, including Sun Solaris and Linux. To compile the source you’ll need (to quote the authors) “a good C++ compiler” as well as the `xforms` library (available from `ftp.via.ecp.fr/pub/xforms`). Although the current version is only 0.10.7, and described as a beta version, it is reasonably complete and usable. The software is written by Matthias Ettrich ably assist by many others, too numerous to mention.

This article was prepared using LyX. While doing so, I only experienced 3 crashes over a period of several hours plus a few oddities which are described.

LyX is a front-end for an existing, working, L^AT_EX 2_ε installation. It won’t work with the older 2.09 release. It likes to use `xdvi` for screen previewing and `dvips` for printing, though other printer drivers may be usable by re-configuring LyX via its configuration file.

To create a WYSIWYG word-processor for L^AT_EX would be an extremely complicated task. So LyX takes a different approach; it uses “.lyx” files, which internally resemble T_EX, but which are not. This allows LyX to manage the ‘style’ of the code creation which, in turn, simplifies what is already a complicated task. [How many different ways are there to make a word bold with L^AT_EX? I can think of at least three, but LyX only understands one.] LyX then ‘interprets’ the code in real-time—it does not constantly run L^AT_EX in order to update the display. Thus the screen shows an approximation to the final output, not the *exact* output. Because LyX uses its own internal format, you cannot use it to open a “.tex” file. Neither should you edit “.lyx” files using `vi` or other text editors.

LyX is not for L^AT_EX ‘power users’. You’ll find many things which are easy using `vi` are just not possible within LyX, as I found while composing this review. However, if you are a L^AT_EX beginner, or you have a requirement to ‘encourage’ users away from inferior and troublesome PC word-processors, LyX may be just what you need.

20 The main Window

The main window resembles any other GUI word processor.³ It has a menu bar along the top plus a single row of icons in the tool bar. At the bottom of the screen a status window gives the user messages about what’s happening. Fig. 1 The main Windowfigure.64 shows the main LyX window.

My first gripe is that the authors have chosen to use the `xforms` library to give the application its look and feel. This is a bit like Motif, but has a number of deviations. Most significant among these is the way the scroll bars work. But once you’ve got over this the look and feel should be acceptable. According to the documentation which comes shipped with LyX, the authors have had to do a lot of work on `xforms` to get it into the current state.

The choice of which tools appear in the tool bar is configurable via the `.lyxrc` file.⁴ Those that are available by default are the only ones which currently have icons, so the choice isn’t that great. This is where I get to air my second gripe—the font selection icons. There are only three; an exclamation mark for *select emphasised style*, I’m happy with that. But the second is a figure of a person (man?) which denotes something called noun style for setting small caps.

³ I’m being careful here not to mention that dreaded word word—damn

⁴Users can have their own or can use a system-wide version

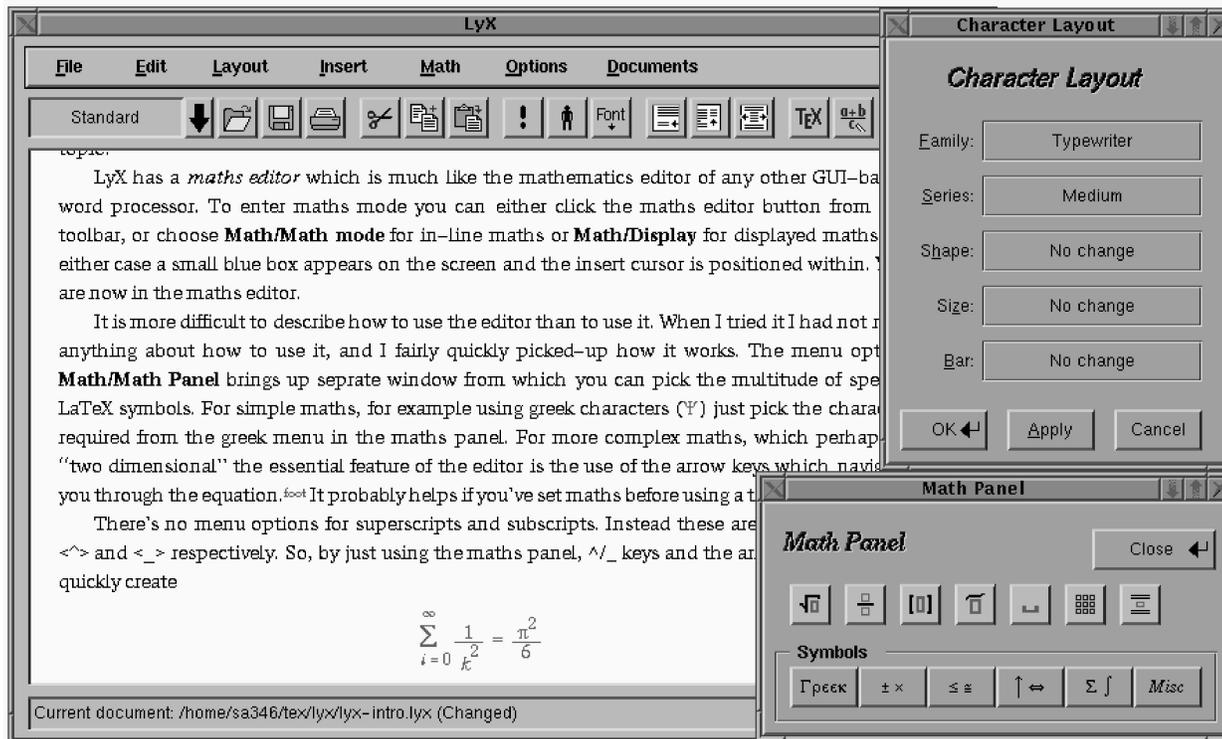


Figure 1. LyX showing the *Character Layout* and *Maths Panels*

Now this *is* weird. The third button has a user-customisable effect, which is set from the **Layout/Character** menu. The menu gets ‘pinned’ to the display so enabling quick changes to family, series, shape and size. Even so, dedicated buttons for typewriter font and bold should be provided in a later release.

On the left of the tool bar is the *style selector*. Here you can choose most of the important L^AT_EX layout styles, such as sections (with and without asterisk) the three basic list environments, verse and quote. There’s also the **LaTeX** option which allows the author to directly enter a L^AT_EX command—these appear in red on the screen. More about this later.

21 Getting started

Choose **New** from the **File** menu. The file picker is definitely *not* Motif, which is one good thing to say about xforms. You can also choose a file template which is used to initialise the document, by inserting its contents into the new one. And away you go!

As you type, LyX automatically breaks the line and wraps the cursor onto the following line. You should only press <Return> to end the paragraph. Using <Control-Return> enters a `\\` command. If you make change to a paragraph, LyX instantly re-formats the screen to show its new appearance. LyX doesn’t hyphenate words, it always breaks at word boundaries. However, remember LyX is only *approximating* to the final appearance. Once you format the document, T_EX will be finding the line breaks instead.

Use <Control-Space> for an unbreakable space (tie) which is shown as a small red □.⁵ New paragraphs are automatically shown indented (but not, of course, immediately following a section). The menu **Layout/Paragraph** can be used to cancel paragraph indentation. This has a global effect on the document.

LyX has intelligent quotes, which are used when the double-quote character is entered. You can select English, French or German quote marks, though the display always shows English.

⁵ I tried, unsuccessfully, to create this symbol by using `\verb*+ +`. This produced a L^AT_EX error, though not when I created a “.tex” file and then formatted that.

Clicking the floppy-disc icon in the tool bar automatically saves your document back to disc. There's also a timed-backup feature.

From the **Edit** menu, there is an "infinite" depth undo facility. LyX also supports PC-style **cut**, **copy** and **paste** via three buttons in the tool bar. It doesn't, however, support the normal X-selection, which is a shame, so you can't export from LyX to other X-applications, though you can import using **Edit/Paste Primary Selection**.

21.1 Changing the Appearance

You have two choices about changing the visual appearance. You can either change the settings *before* you type something, or you can go back, highlight it and then apply some new combination of effects. For many commands 'sensible' things happen when attributes are changed and nothing is highlight. For example if I were to pick left-block from the **Layout/Paragraph** menu in the middle of this paragraph, the new setting would be automatically applied to the whole of the current paragraph.

22 Lists

The easiest way to create a list is to enter each item as a separate paragraph, then highlight all the items and choose the required list type from the style drop-down. When a description list is used, the first word of the item is automatically emboldened and you must use unbreakable spaces (`<Control-Space>`) to include more than one word.

LyX provides a forth list-type, **list**, which is based on the \LaTeX `list` environment. An example follows;

itemized	produces bullet-points
enumerated	produces numbered lists
description	produces a bold subject followed by a description
list	produces this type of list. Note that the width used for each label is the same. The actual width used is set by simply highlighting the entire list and setting the label width using the Layout/Paragraph menu, then clicking Apply .

22.1 Lists within lists

The tool bar provides the button *change environment depth*. This is used to insert a new list 'level' into an existing list. For example to achieve;

- This is an item of an `itemize` list
 - This is the first item of a list *within* the first
- end.

the *change environment depth* was used after entering the first item so that the following item was at the next level. After entering the second line, `<Return>` puts LyX back at the previous level. Using the highlight-then-apply model, it's possible to select a number of items for 'moving' to a new level.

1. This is an item of an enumerated list
 - When typed this was item 2, it will be moved to a new level
 - When typed this was item 3
2. When typed this was item 4

23 Footnotes and marginal notes

A footnote is inserted into the document by using the dedicated button in the tool bar. This opens a text box on the screen into which the footnote text is entered. By clicking-left on the box's *foot* tab, the footnote is closed and a small red "foot" appears in the main text. Double-clicking left on this, re-opens the box for editing. Clicking-right in the opened box allows the footnote to be deleted (called *melting*). Once the footnote has been melted, its text just becomes part of the document. The opposite is true as well, just highlight a piece of text, click the footnote button, and voilà—one footnote.

Marginal notes are handled in an identical manner.

24 Figures and tables

Figures and tables can be inserted either "here" or as floats. You appear to have to choose at the time these are created, you don't seem to be able to change your mind later. Inserting a table causes it to appear on the screen centred under the "current" sentence. Choosing a table float opens up a red table box into which the table can be inserted and a caption added to the **Table:** prompt. The box can then be collapsed in a same way as for footnotes.

24.1 The Table Editor

Creating a table (actually a `tabular` environment) is achieved using a *table editor*. The first stage is to select the number of rows and columns. Sliders are provided for this, defaulting to 5x5. Once the size is set, an empty table is inserted into the document, with the cursor positioned in the top left cell. The table's initial layout assumes the first row will be used as a heading and it is separated from the rest of the table by two horizontal lines (`\hlines`). Rows and columns have separating rules.

With the cursor positioned in a cell, entering text causes the text to appear in the cell, and all cells of the same column automatically expand to the same width—exactly as you'd expect with the `tabular` environment. To edit neighbouring cells, simply use the cursor arrow keys, or click the left mouse button.

With the insert-cursor in a cell, clicking-right displays a table menu which allows you to change the following;

- the text alignment from **center** to **left** or **right**
- delete the column or row
- append a new column or row
- choose the rules used separate cells (select rule **left**, **top** and/or **bottom**). If no rule is selected, a dotted-line is drawn to show the separation between rows or columns.
- combine adjacent cells into a single cell (*multicolumn*). This is achieved by high-lighting the cells and choosing **multicolumn** from the table-menu. I found that in version 0.10.7, this sometimes caused problems.

Using the table editor, this is one of the examples from *Lampport*;

<i>type</i>	<i>style</i>	
smart	red	short
rather silly	puce	tall

In *Lampport* the table had a double rule at the top and bottom. This particular feature seems not to be possible with the table editor. `\cline`'s aren't possible either. Another tabular feature which I almost always use, namely `\arraystretch`, must be set by entering the required command directly (see below).

Vertical space above and/or below the table can be added by high-lighting the entire table and then changing the paragraph settings from the **Layout/Paragraph** menu. This also permits the table alignment to be changed to left of right.

24.2 Figures

LyX allows the inclusion of figures in one of three formats;

encapsulated PostScript which is displayed centred on the page

in-lined encapsulated PostScript which is displayed aligned with the adjacent text

LaTeX which permits you to `\input` a file and display it centred on the page

The first two formats create boxes on the screen with an **F** to signify a figure. Double-clicking-left on the box opens up a dialogue box in which the details of the figure are programmed. Apart from the obvious one of filename, the author can choose the size, rotation angle and a command which LyX uses to display the image. (It defaults to using **ghostscript**.) Once the parameters have been entered and applied, LyX updates the display with the image.

25 Setting Mathematics

LyX has a *maths editor* which is much like the mathematics editor of any other GUI-based word processor. To enter maths mode you can either click the maths editor button from the tool bar, or choose **Math/Math mode** for in-line maths or **Math/Display** for displayed maths. In either case a small blue box appears on the screen and the insert cursor is positioned within. You are now in the maths editor.

It is more difficult to describe how to use the editor than to use it—and even I quickly picked-up how it works. The menu option **Math/Math Panel** brings up a separate window from from which you can pick the multitude of special LaTeX symbols. For simple maths, for example using Greek characters, just pick the character required from the menu in the maths panel. For more complex maths, which perhaps requires a “two dimensional” layout, the essential feature of the editor is the use of the arrow keys which navigate you through the equation.

There are no menu options for superscripts and subscripts which greatly simplifies the maths panel. Instead, these

are entered by using $\langle \wedge \rangle$ and $\langle _ \rangle$ respectively. So, by just using the maths panel, $\wedge / _$ keys and the arrow keys, you can quickly create

$$\sum_{i=0}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

Or, how about

$$y = \int_0^{\infty} \left\{ \frac{\sin^2 \left(\alpha \sqrt{\frac{ax+b}{cx+d}} \right)}{\sqrt{\beta + \alpha x}} \right\}^{\frac{a+b}{c+d}} dx$$

The appearance of the maths on the screen is more than adequate to show the detail of what has been selected. The Greek characters and special symbols are very well drawn.

The maths editor supports roots, fractions, delimiters, “over symbols” (called decorations), arrays and maths spacing. It does not appear to support the non-roman maths fonts, log-like functions, nor can I determine how to get numbered equations. Neither can you swap a piece of maths between ‘display’ and ‘in-line’; you have to choose at the point of creation.

26 Document styles

From the **Layout/Document** menu you can choose a number of parameters which affect the whole document. These include the document class, the typeface (L_YX supports the common POSTSCRIPT faces) body type size (10pt, 11pt etc), paper size, paper orientation (landscape or portrait), single or two-sided, single or double column,⁶ language (from which you can choose from about 20 possibilities), baseline stretch, section numbering depth and table-of-contents depth. Also from the **Layout** menu, the **Preamble** option allows you to enter a series of lines which are added verbatim to the document’s preamble.

There’s good support for letters with L_YX; you can select **letter** as the document class and the standard installation comes with a suitable template. Selecting the **letter** class also changes the L^AT_EX formatting options available from the style drop-down. Included is **cc**, **Signature**, **Opening** and **Closing**.

27 Entering L^AT_EX Commands

Sooner or later you will want to achieve some formatting effect which L_YX cannot currently handle. To do this you can enter a T_EX or L^AT_EX command directly into the document. You can do this in a number of ways, the easiest of which is to use the dedicated button in the tool bar (with the T_EX logo). Clicking this puts L_YX into “T_EX mode” and what you type is shown in red. Effectively what you type is added to the file verbatim, and no interpretation is undertaken.

While creating this document I have had used this feature only a few times. However, it will be clear by now that quite significant amounts of L^AT_EX are not yet covered by L_YX; rules, boxes, the minipage environment and the tabbing environment, to name a few. And then there’s all those packages which are available.

28 Interfacing with L^AT_EX

One of the most frustrating features with teaching beginners L^AT_EX is explaining how to deal with the inevitable errors which result from formatting. L_YX solves this difficult problem in a way which must have taken a great deal of work. Whenever L_YX runs L^AT_EX for you, any errors which occur are flagged by inserting a red, Error into the document at the point where the error occurred. By clicking-left over the box, it opens to show you the description of the error. This is really neat. From the **Edit** menu, use **Goto error** to rapidly move between the different errors.

Printing is possible from the **File** menu as well as from a dedicated button in the tool bar. A print dialogue allows the author to choose the name of the printer (that which would normally be appended to a `-P` option using an **lpr** command), the stacking order (normal or reverse) and whether to print to a file or printer. When the document is printed, L_YX doesn’t bother to re-format the document if nothing has been changed. Because L^AT_EX is only run once this causes problems if your document uses cross-references. The work-around is to format the document twice using **File/Run LaTeX** and then to use the print button.

⁶ though L_YX does not display two columns if double column is chosen

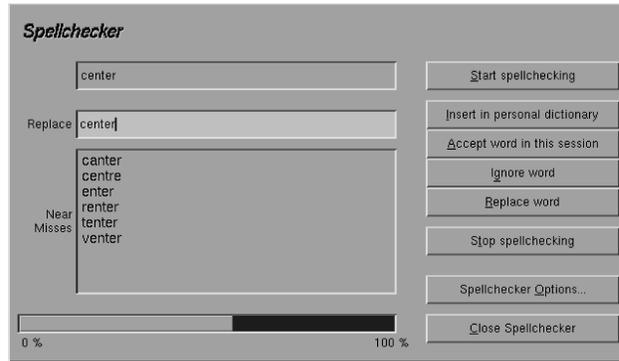


Figure 2. The spell checker window

It’s possible to create (to quote the menu) a “nice” `.tex` file from the “`.lyx`” file. This can be sent to someone who doesn’t use LyX, and will, naturally, produce an identical output, notwithstanding the usual issues of fonts and packages. This is how this document was transmitted to the editors of *Baskerville*.

29 Spell Checker

LyX supports a spell checker by interfacing with the GNU **ispell** command, which must be separately installed. Using the spell checker is simplicity itself. Mis-spelled words are highlighted in the main document, while in a separate window the alternate choices are displayed, together with a number of control buttons (see Fig. 2Spell Checkerfigure.83). A progress bar is also provided.

30 A Final Word

There are a number of \LaTeX features which are supported by LyX but which have not been covered in this review. These include tables-of-content, lists-of-figures and tables, cross references, bibliographic citations, verse, various commands for building title pages (title, author etc.).

I’m a die-very-hard user of **vi**, but even after a couple of hours with LyX, I now find myself wishing for word-wrap and point-click-type when using an xterm. There’s no getting away from it, GUI applications are appealing.

Its difficult for me to over-express my admiration for this program and my thanks to the authors for their effort in producing it. It’s brilliant! Even the “beta” 0.10.7 version has proved to be extremely reliable—I regard 3 crashes as well within acceptable limits. Such an editor is, in my view, long overdue for the \LaTeX world and the fact that Unix has been chosen as the first delivery OS heartens me no-end. Well done Matthias and the rest of the team!

31 Typesetting the L^AT_EX Logo: a Saga

Robin Fairbairns

When I was collating the first edition of the UKTUG FAQ, one of the people I asked to review it was Barbara Beeton, and one of her comments was that the L^AT_EX logo looked awful, specifically when typeset in *Baskerville Italic*.

Barbara's observation set me on a quest for a resilient macro to typeset this curious object; I still haven't found any philosopher's stone for this task, but I have at least got a formula that enables one to codify some working parameters in a L^AT_EX-like sort of way

Sebastian Raatz, in one of his fulminating editorials, cried 'a pox on all these logos'. In retrospect one could wish that Lamport hadn't taken us beyond Knuth's original T_EX logo (which seemed amusing at first encounter), but while one can steadfastly ignore most other logos, it seems difficult to ignore L^AT_EX (whether you're a convinced user or a virulent hater). I hope that the work I describe here will make its own small contribution to the beauty of descriptions, in L^AT_EX, of L^AT_EX itself . . . or will, at least, draw them back from the abyss of hideousness that the logo itself seems inexorably to pull towards.

32 The Old L^AT_EX coding

The last release of 2.09 (dating from March 1992), had:

```
\def\p@LaTeX{\reset@font\rm L\kern-.36em%
  \raise.3ex\hbox{\sc a}\kern-.15em%
  T\kern-.1667em\lower.7ex\hbox{E}\kern-.125emX}}
\def\LaTeX{\protect\p@LaTeX}
```

Apart from the `\reset@font`, this is (in essence) Lamport's original coding; the definition of `\p@LaTeX` provides the basis for the robust command `\LaTeX` (Lamport's original was `fragile`).

The output the macro produces is reasonable, *if* you're using `cmr` at a 'reasonable' size: at 10pt, we see L^AT_EX. It's odd that its code includes that for (Knuth's) logo `\TeX` (rather than referring to the older macro), and it's noteworthy that neither it nor Knuth's code makes the spacefactor adjustment due at the end of the word⁷ (in English typesetting, it's common practice to set the capitals in acronyms and logos as if they were lower-case letters).

The problems that arise from the macro are:

- that it steadfastly sets in the default font (`cmr`), and
- that its use of `\sc` restricts it to a smaller range of sizes than one might hope.

These restrictions make it appear odd in section titles (which are set in bold in L^AT_EX's standard classes), and prevent (what seems to me) entirely reasonable use in sans fonts, etc. However, since it's now history, I shan't discuss it further.

33 The standard L^AT_EX coding

I don't any longer have the sources, but as I recall at least some of the pre-release versions of L^AT_EX 2_ε had something similar to the L^AT_EX 2.09 version of the logo. This version, instead of using the 'old' font command `\sc` (which selects some 'fixed' font, rather than a font related to that which surrounds it), used the L^AT_EX command `\scshape`. (The same would also have been true of pre-L^AT_EX 2_ε versions of the NFSS with their subversion of the L^AT_EX 2.09 font commands.) This had the unfortunate effect that L^AT_EX warned you about using `\LaTeX` in a section title, since almost no font family defines a bold-series Caps-and-Small-Caps font

The appearance of the present L^AT_EX logo was therefore a minor relief for navel-contemplating⁸ users such as myself. The new logo coding sets the 'A' of L^AT_EX in the scriptfont size related to the current text size, and with its apex at the same level as the top of the 'T'. This logo actually looks slightly different from the one in L^AT_EX 2.09, but is noticeably more stable:

```
\DeclareRobustCommand{\LaTeX}{L\kern-.36em
  {%
```

⁷This observation is due to Barbara Beeton.

⁸'Real' users, writing about anything other than L^AT_EX itself would never have noticed the problem!

Typesetting the L^AT_EX Logo: a Saga

```
\sbox\z@ T%
\ vbox to\ht0{\hbox{\$ \m@th$%
  \csname S@\f@size\endcsname
  \fontsize\sf@size\z@
  \math@fontsfalse\selectfont
  A}%
  \vss}%
}%
\kern-.15em
\TeX}
```

As you can see, the code sets ‘T’ into a scratch `\hbox` which it then uses as the height of a `\vbox` into the top of which the ‘A’ is stuck. The size of the ‘A’ is selected in the curious sequence involving an empty maths mode (with zero `\mathsurround`), the setting-up of current maths sizes using `\S@\langle f@size \rangle` (as it were), and then using the `\sf@size` (which is the ‘current script font size’) thus defined.

This version of the logo deals with some of the problems of the L^AT_EX209 version, and pushes back the boundaries of difficulty into new areas:

- very small font sizes, where the scriptfont size doesn’t differ much (or at all) from the textfont size,
- font sizes for which no `\DeclareMathSizes` has been issued: in this case, L^AT_EX will calculate a value for the size of `\scriptstyle` which ought to be appropriate, but which could well not be available within the set of fonts L^AT_EX is willing to employ,
- thoroughly peculiar uses (for example, the man who complained on `comp.text.tex` that the `textfit` package produces odd effects, like: , which fails because the font is selected outside of the NFSS, and so the ‘A’ matches whatever size is NFSS-selected — in this case 10pt⁹), and
- use in fonts that don’t match the kerning parameters used.

It’s the last problem that I’ve been attempting to address in my investigations of the logo.

34 My first attempt

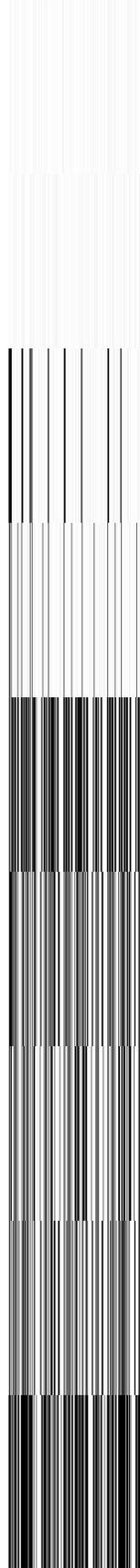
This code deals with the slant of an italic character, and has been used in the published editions of the FAQ.

```
\DeclareRobustCommand{\LaTeX}{L%
  {%
    \setbox0\hbox{T}%
    \setbox\@tempboxa\hbox{\$ \m@th$%
      \csname S@\f@size\endcsname
      \fontsize\sf@size\z@
      \math@fontsfalse\selectfont
      A}%
    \@tempdima\ht0
    \advance\@tempdima-\ht\@tempboxa
```

That sets up the amount by which the ‘A’ will be raised, in `\@tempdima`; the next line of code extracts the font dimension that gives the slant, and multiplies `\@tempdima` by it; this is simple coordinate geometry, and gives an amount by which to adjust the negative kern in the standard version of the logo. We then calculate an actual value to use for the kern, and complete the fractured coding of the standard logo.

```
\@tempdima
\strip@pt\fontdimen1\font\@tempdima
\advance\@tempdima-.36em
\kern\@tempdima
\ vbox to\ht0{\box\@tempboxa
  \vss}%
}%
```

⁹That example was achieved by selecting `\huge` outside of the NFSS: the original complaint was about a 1 inch high logo, at which size it’s difficult to see that there’s an ‘A’ there at all!



```
\kern-.15em
\TeX}
```

This code improves the appearance of inclined copies of the logo — inclined copies are absolutely dire if you’re using Baskerville fonts (as we do in *Baskerville*): compare \LaTeX (the standard logo) with \LaTeX (the above code).

35 But the standard code doesn’t even work with CM-family fonts!

Section titles in TUGboat are set in `cmsstbx`¹⁰: the fragility of the logo is demonstrated by the fact that the standard code fails with this font: \LaTeX . When the first ‘ \LaTeX macros’ section turned up since I joined the TUGboat production team, Barbara Beeton asked me to ‘sort it out’. Since this problem is occurring for an upright font, the trick I developed for use in *Baskerville* doesn’t help.

I could have taken a hint from the documentation that’s included in the \LaTeX distribution, which defines a different logo for each font family one might use to print it; however, I didn’t notice this, and developed a version that can in principle behave differently for every value of font family, series and shape.

The `\LaTeX` command itself is simple: it simply chooses a set of pre-defined parameters (or, in default of that, selects the parameters used in the standard logo), and then relays to a patched version of the standard logo:

```
\DeclareRobustCommand\LaTeX{%
\expandafter\let\expandafter\@tempa
\csname @LaTeX@\f@family/\f@series/%
  \f@shape\endcsname
\ifx\@tempa\relax\let\@tempa\@LaTeX@default\fi
\expandafter\@LaTeX\@tempa}
```

The patched version of the standard logo is equally trivial:

```
\newcommand\@LaTeX[2]{L\kern-#1em
{%
  \sbox\z@ T%
  \vbox to\ht0{\hbox{\$m@th$%
    \csname S@\f@size\endcsname
    \fontsize\sf@size\z@
    \math@fontsfalse\selectfont
    A}%
  \vss}%
}%
\kern-#2em%
\TeX}
```

The parameters are defined by the command

```
\newcommand\DeclareLaTeXLogo[5]{%
\expandafter\def
\csname @LaTeX@#1/#2/#3\endcsname
{#{#4}{#5}}}
```

whose parameters are font family, series and shape, and then the two extra parameters to be used by `\@LaTeX`. The default setting is simply:

```
\def\@LaTeX@default{ {.36}{.15}}
```

My `ltugboat` class defines three sets of CM-family parameters:

```
\DeclareLaTeXLogo{cmsstbx}{n}{.3}{.15}
\DeclareLaTeXLogo{cmr}{m}{it}{.3}{.27}
\DeclareLaTeXLogo{cmr}{bx}{it}{.3}{.27}
```

the first of which deals with the problem that Barbara Beeton had highlighted.

Finding sets of parameters to use isn’t entirely trivial; I generate a set of logos with different parameters for the font

¹⁰TUGboat ‘sections’ are collections of related papers within one issue; sections within a TUGboat paper are like sections within any \LaTeX document.

family, series and shape under consideration, examine them, adjust the set of parameters and iterate until I have something that looks satisfactory. The last sample in this exercise for Baskerville medium, normal, is shown in figure 1The standard \LaTeX codingfigure.91.

36 Conclusion: future work

The macros I describe above are available on CTAN in directory `macros/latex/contrib/supported/ltx-logo` (together with the source of this note, by way of justification). The `ltx-logo` package takes font family names as package options: for each option quoted, the package will define any logo parameter sets it knows for that family: at the time of writing, the package knows about the `cm` family (using the parameters shown above for `cms`/`bx`/`n`, `cmr`/`m`/`it` and `cmr`/`bx`/`it`) and the Monotype Baskerville families `mbv` and `mbvx` (the latter includes use of the so-called ‘expert’ fonts: for the purposes of the \LaTeX logo, the required parameters are the same as for the non-expert set).

I remarked above that the \LaTeX team had defined special-purpose \LaTeX logos for a number of font families, in the macros used for the documentation distributed with \LaTeX . Their approach (which was developed independently of mine) isn’t as general as mine, but I was interested to note that they also redefine the \TeX logo: I hadn’t bothered with that, on the grounds that it seemed ‘mostly OK’ in texts I had set. I suppose it offers an avenue for still further enthralling developments in my study of \TeX -based logos.

The macros on CTAN include the test code I used to evaluate parameters for the fonts. People who feel moved to work out sets for their own favourite font family are encouraged to submit them to me for inclusion in the distribution

....

37 The natbib package

Patrick W. Daly
Max-Planck-Institut für Aeronomie
Max-Planck-Str. 2
D-37191 Katlenburg-Lindau
Germany
E-mail: daly@linmpi.mpg.de

Summary

The `natbib` package is a reimplementation of the \LaTeX `\cite` command, to work with both author–year and numerical citations. It is compatible with the standard bibliographic style files, such as `plain.bst`, as well as with those for `harvard`, `apalike`, `chicago`, `astron`, `authordate`, and of course `natbib`.

In contrast to the packages listed above, the `natbib` package supports not only the various author–year bibliography styles, but also those for standard numerical citations. In fact, it can also produce numerical citations even with an author–year bibliographic style, something that permits easy switching between the two citation modes. To this end, replacements for the standard \LaTeX `.bst` files are also provided.

It is possible to define the citation *style* (type of brackets and punctuation between citations) and even to associate it with the name of the bibliographic style so that it is automatically activated. Citation styles can be defined for local `.bst` files by means of a configuration file `natbib.cfg`.

It is compatible with the packages: `babel`, `index`, `showkeys`, `chapterbib`, `hyperref`, `koma` and with the classes `amsbook` and `amsart`. It can also emulate the sorting and compressing functions of the `cite` package (with which it is otherwise incompatible).

The `natbib` package therefore acts as a single, flexible interface for most of the available bibliographic styles.

38 Introduction

The first problem of using author–year literature citations with standard \LaTeX is that the two forms of citations are not supported. These are:

textual: ... as shown by Jones et al. (1990) ...
parenthetical: ... as shown (Jones et al., 1990) ...

There is only one `\cite` command to do both jobs.

A second problem is that the `thebibliography` environment for listing the references insists on including the *labels* in the list. These labels are normally the numbers, needed for referencing. In the author–year system, they are superfluous and should be left off. Thus, if one were to make up a bibliography with the author–year as label, as

```
\begin{thebibliography}{...}  
\bibitem[Jones et al., 1990]{jon90}  
Jones, P. K., . . .  
\end{thebibliography}
```

then `\cite{jon90}` produces the parenthetical citation [Jones et al., 1990], but there is no way to get the textual citation. Furthermore, the citation text will also be included in the list of references.

The final problem is to find a `BIBTEX` bibliography style that will be suitable.

39 Previous solutions

Although the author–year citation mode is not supported by *standard* \LaTeX , there are a number of contributed packages that try to solve this problem. The various bibliographic styles (`.bst` files) that exist are usually tailored to be used with a particular \LaTeX package.

I have found a large number of `.bst` files on file servers that may act as indicators of the various systems available.

39.1 *The natsci.bst style*

What gave me my first inspiration was Stephen Gildea's `natsci.bst` for use with his `agujgr.sty` file. This showed me that the problem was solvable. However, Gildea's style formats `\bibitem` just as I illustrated above: with an optional label consisting of abbreviated authors and year. Thus only parenthetical citations can be accommodated. The list of references, however, is fixed up in his style files.

39.2 *The apalike.bst style*

Oren Patashnik, the originator of \LaTeX and the standard `.bst` files, has also worked on an author-year style, called `apalike.bst` with a corresponding `apalike.sty` to support it. Again, only the parenthetical citation is provided. Except for the fact that his style works with version 0.99 of \LaTeX , its functionality is identical to that of the `natsci` files.

Patashnik does not like author-year citations. He makes this very clear in his \LaTeX manuals and in the header to `apalike.bst`. Nevertheless, one should respect his work in this area, simply because he should be the best expert on matters of \LaTeX . Thus `apalike.bst` could be the basis for other styles.

The form of the `thebibliography` entries in this system is

```
\bibitem[Jones et al., 1990]{jon90}...
```

as illustrated earlier. This is the most minimal form that can be given. I name it the `apalike` variant, after Patashnik's `apalike.bst` and `apalike.sty`. However, there could be many independent `.bst` files that follow this line.

The bibliography style files belonging to this group include: `apalike`, `apalike2`, `cea`, `cell`, `jmb`, `phapalik`, `phppcf`, and `phrmp`.

39.3 *The newapa style*

A major improvement was achieved with `newapa.bst` and the accompanying `newapa.sty` files by Stephen N. Spencer and Young U. Ryu. Under their system, three separate items of information are included in the `\bibitem` label, to be used as required. These are: the full author list, the abbreviated list, and the year. This is accomplished by means of a `\citeauthoryear` command included in the label, as

```
\bibitem[\protect\citeauthoryear{Jones, Barker,
and Williams}{Jones et al.}{1990}]{jon90}...
```

Actually, this only illustrates the basic structure of `\citeauthoryear`; the `newapa` files go even further to replace some words and punctuation with commands. For example, the word 'and' above is really `\betweenauthors`, something that must be defined in the `.sty` file. Of course, `\citeauthoryear` is also defined in that file. A number of different `\cite` commands are available to print out the citation with complete author list, with the short list, with or without the date, the textual or parenthetical form.

Thus the `\citeauthoryear` entry in `\bibitem` is very flexible, permitting the style file to generate every citation form that one might want. It is used by a number of other styles, with corresponding `.sty` files. They all appear to have been inspired by `newapa.bst`, although they lack the extra punctuation commands.

Bibliographic style files belonging to the `newapa` group include `newapa`, `chicago`, `chicagoa`, `jas99`, and `named`. The last of these, `named.bst`, uses `\citeauthoryear` in a slightly different manner, with only two arguments: the short list and year.

39.4 *The Harvard family*

The same effect is achieved by a different approach in the Harvard family of bibliographic styles. Here a substitute for `\bibitem` is used, as

```
\harvarditem[Jones et al.]{Jones, Baker, and
Williams}{1990}{jon90}...
```

The accompanying interface package file is called `harvard.sty` and is written by Peter Williams and Thorsten Schnier. It defines `\harvarditem` as well as the citation commands `\cite`, for parenthetical, and `\citeasnoun`, for textual citations. The first citation uses the long author list, following ones the shorter list, if it has been given in the optional argument to `\harvarditem`.

Bibliography styles belonging to the Harvard family are `agsm`, `dcu`, `kluwer`.

This package has been updated for \LaTeX 2\epsilon , with many additions to add flexibility. The result is a powerful interface that should meet most citation needs. (It does not suppress repeated authors, though, as `natbib` does.)

The natbib package

39.5 The astronomy style

Apparently realizing the limitations of his apalike system, Oren Patashnik went on to develop a ‘true’ apa bibliographic style, making use of the method already employed by an astronomy journal. This is actually very similar to the newapa label but with only the short list of authors:

```
\bibitem[\protect\astroncite  
{Jones et al.}{1990}]{jon90} ...
```

It requires the package file `astron.sty` or any other style that defines `\astroncite` appropriately.

Bibliographic styles belonging to the astronomy group are `apa`, `astron`, `bbs`, `cbe`, `humanbio`, `humannat`, and `jtb`.

This is as good as the `\citeauthoryear` command, although not as flexible since the full list of authors is missing.

39.6 The authordate style

Finally, I have also found some packages making use of a label command called `\citename` in the form

```
\bibitem[\protect\citename  
{Jones et al., }1990]{jon90} ...
```

This is not a good system since the author list and date are not cleanly separated as individual arguments, and since the punctuation is included in the label text. It is better to keep the punctuation fully removed, as part of the definitions in the `.sty` file, for complete flexibility.

Bibliographic styles belonging to this group are `authordate1`, `authordate2`, `authordate3`, `authordate4`, and `aaai`-named, with an accompanying style file `authordate1-4.sty`.

40 The natbib system

The form of the `\bibitem` entry that I have used for all my bibliographic styles is only slightly more complicated than the minimal one, but allows a clean separation between authors and date:

```
\bibitem{Jones et al.(1990)}{jon90}...
```

or alternatively

```
\bibitem{Jones et al.(1990)Jones, Baker,  
and Williams}{jon90}...
```

One weakness of the `natbib` format is that it fails if the author list itself contains parentheses! This may be fixed by grouping the author list in curly braces.

I wanted to name the system something like ‘natural sciences bibliography’, intending it to be a variant of `natsci.sty`. Since that name was already taken, I resorted to the rather cryptic, and definitely ugly, `natbib`.

The `natbib.sty` package¹¹ supports not only my own `\bibitem` format, but also all the others described above, plus numerical citation modes. The additional questions of citation style (type of brackets, commas or semi-colons between citations) can be defined once and for each `.bst` file and need never be specified explicitly in the source text. The use of `\cite` is the same for all citation styles, meaning that the additional features that might be available in the ‘proper’ `.sty` file will be missing. (This could be changed later.) The result is a single \LaTeX package to handle *all* the bibliographic styles in a uniform manner.

In previous versions of this package, each `.bst` file was either numerical or author–year only. Applying the wrong mode led to grotesque results, and the mode could not always be selected automatically. With version 6.0 onwards, all the author–year bibliographic style files can also be used for *numerical* citations, by simply selecting the mode in one of the ways described on pp. 38 and 35. It is not possible to employ author-year citations with pure numerical `.bst` files, and never will be.

41 Using this package

`natbib` is loaded in the normal way:

```
\documentclass[...]{...}  
\usepackage[options]{natbib}
```

The options available provide a means of specifying the punctuation for citations:

¹¹Formerly called a *style file* in the older \LaTeX 2.09 terminology.

round (default) for round parentheses;
square for square brackets;
curly for curly braces;
angle for angle brackets;
colon (default) to separate multiple citations with colons;
comma to use commas as separators;
authoryear (default) for author–year citations;
numbers for numerical citations;
super for superscripted numerical citations, as in *Nature*;
sort orders multiple citations into the sequence in which they appear in the list of references;
sort&compress as **sort** but in addition multiple numerical citations are compressed if possible (as 3–6, 15);
longnamesfirst makes the first citation of any reference the equivalent of the starred variant (full author list) and subsequent citations normal (abbreviated list);
sectionbib redefines `\thebibliography` to issue `\section*` instead of `\chapter*`; valid only for classes with a `\chapter` command; to be used with the `chapterbib` package.

If any of the formatting options are selected, the predefined citation styles in the commands `\bibstyle@bst` will be no longer be effective. If either `\bibpunct` or `\citestyle` is given in the preamble, the above punctuation options will no longer hold.

Basic commands

In this paper, I distinguish between the citation *mode* (author–year or numerical) and citation *style* (the type of punctuation used for citations). The citation style is something that is independent of the bibliography style and is not programmed in the `.bst` files.

41.1 Basic citation commands

The `natbib` package has two basic citation commands, `\citet` and `\citep` for *textual* and *parenthetical* citations, respectively. There also exist the starred versions `\citet*` and `\citep*` that print the full author list, and not just the abbreviated one. All of these may take one or two optional arguments to add some text before and after the citation.

```
\citet{jon90}
⇒ Jones et al. (1990)
\citet[chap.~2]{jon90}
⇒ Jones et al. (1990, chap. 2)
\citep{jon90}
⇒ (Jones et al., 1990)
\citep[chap.~2]{jon90}
⇒ (Jones et al., 1990, chap. 2)
\citep[see][ ]{jon90}
⇒ (see Jones et al., 1990)
\citep[see][chap.~2]{jon90}
⇒ (see Jones et al., 1990, chap. 2)
\citet*{jon90}
⇒ Jones, Baker, and Williams (1990)
\citep*{jon90}
⇒ (Jones, Baker, and Williams, 1990)
```

The starred versions can only list the full authors if the `.bst` file supports this feature; otherwise, the abbreviated list is printed.

In standard \LaTeX , the `\cite` command can only take a single optional text for a note after the citation; here, a single optional text is a post-note, while two are the pre- and post-notes. To have only a pre-note, it is necessary to provide an empty post-note text, as shown above.

Multiple citations

Multiple citations may be made as usual, by including more than one citation key in the `\cite` command argument. *If adjacent citations have the same author designation but different years, then the author names are not reprinted.*

```
\citet{jon90, jam91}
⇒ Jones et al. (1990); James et al. (1991)
```

The natbib package

```
\citep{jon90,jam91}
⇒ (Jones et al., 1990; James et al. 1991)
\citep{jon90,jon91}
⇒ (Jones et al., 1990, 1991)
\citep{jon90a,jon90b}
⇒ (Jones et al., 1990a,b)
```

Numerical mode

These examples are for author–year citation mode. In numerical mode, the results are different.

```
\citet{jon90}
⇒ Jones et al. [21]
\citet[chap.~2]{jon90}
⇒ Jones et al. [21, chap. 2]
\citep{jon90}
⇒ [21]
\citep[chap.~2]{jon90}
⇒ [21, chap. 2]
\citep[see][ ]{jon90}
⇒ [see 21]
\citep[see][chap.~2]{jon90}
⇒ [see 21, chap. 2]
\citep{jon90a,jon90b}
⇒ [21, 32]
```

The authors can only be listed if the .bst file supports author–year citations. The standard .bst files, such as plain.bst, are numerical only and transfer no author–year information to L^AT_EX. In this case, \citet prints “(author?) [21].”

In the original versions of natbib, the traditional \cite command was used for both textual and parenthetical citations. The presence of an empty optional text in square brackets signalled parenthetical. This syntax has been retained for compatibility, but is no longer encouraged. This means that \cite (without notes) is the same as \citet in author–year mode, whereas in numerical mode, it is the same as \citep. The starred version, as well as the one or two optional notes, may also be used.

It is possible to have multiple citations sorted into the same sequence as they appear in the list of references, regardless of their order as arguments to the \cite commands. The option sort is required for this feature. See p. 41.

Some publishers require that the first citation of any given reference be given with the full author list, but that all subsequent ones with the abbreviated list. Include the option longnamesfirst to enable this for natbib. See p. 41.

41.2 Extended citation commands

As an alternative form of citation, \citealt is the same as \citet but *without any parentheses*. Similarly, \citealp is \citep with the parentheses turned off. Multiple references, notes, and the starred variants also exist.

```
\citealt{jon90}
⇒ Jones et al. 1990
\citealt*{jon90}
⇒ Jones, Baker, and Williams 1990
\citealp{jon90}
⇒ Jones et al., 1990
\citealp*{jon90}
⇒ Jones, Baker, and Williams, 1990
\citealp{jon90,jam91}
⇒ Jones et al., 1990; James et al., 1991
\citealp[pg.~32]{jon90}
⇒ Jones et al., 1990, pg. 32
```

Partial citations

In author–year schemes, it is sometimes desirable to be able to refer to the authors without the year, or vice versa. This is provided with the extra commands

```
\citeauthor{jon90}
  ⇒ Jones et al.
\citeauthor*{jon90}
  ⇒ Jones, Baker, and Williams
\citeyear{jon90}
  ⇒ 1990
\citeyearpar{jon90}
  ⇒ (1990)
```

There also exists a command `\citefullauthor` which is equivalent to `\citeauthor*`. If the full author information is missing, then `\citeauthor*` is the same as `\citeauthor`, printing only the abbreviated list. This also applies to the starred versions of `\citet` and `\citep`. If the author or year information is missing (as is the case with the standard L^AT_EX .bst files), these commands issue a warning. They may also be used with numerical citations, provided an author–year .bst file is being employed.

All `\cite..` commands have the same syntax, allowing multiple citations and up to two notes (there is, however, no starred `\citeyear` variant). It does not really make much sense to add notes to `\citeyear` and `\citeauthor`, especially with multiple citations; however, this can be done — there will be no error message — but the results are sometimes strange. For example, in numerical mode, the notes are fully ignored, while in author–year mode, only the post-note is accepted. Multiple citations in `\citet` are also not recommended (nor are they in my opinion meaningful), but if they are used with notes, the pre-note will appear before each year, and the post-note only after the last year. These are admittedly bugs, but the effort to remove them is not justified by the questionable usefulness of these features.

In summary, notes are only intended for `\citep` but they may also be used with `\citet` in author–year mode, with single citations. In any other situation, the results are unpredictable.

41.3 Selecting citation punctuation

The above examples have been printed with the default citation style. It is possible to change this, as well as to select numerical or author–year mode, by means of the `\bibpunct` command, which takes one optional and six mandatory arguments. The six mandatory arguments:

1. the opening bracket symbol, default = (
2. the closing bracket symbol, default =)
3. the punctuation between multiple citations, default = ;
4. the letter ‘n’ for numerical style, or ‘s’ for numerical superscript style, any other letter for author–year, default = author–year; note, it is not necessary to specify which author–year interface is being used, for all will be recognized;
5. the punctuation that comes between the author names and the year (parenthetical case only), default = ,
6. the punctuation that comes between years or numbers when common author lists are suppressed (default = .); if both authors and years are common, the citation is printed as ‘1994a,b’, but if a space is wanted between the extra letters, then include the space in the argument, as { , ~}.

For numerical mode, `\citet{jon90,jon91}` produces ‘Jones et al. [21, 22]’ with this punctuation between the numbers. A space is automatically included for numbers, but not for superscripts.

The optional argument is the character preceding a post-note, and default is a comma. The `\bibpunct` command must be issued in the preamble, that is, before `\begin{document}`.

Example 1, `\bibpunct[[]]{}{,}{a}{}{;}` changes the output of

```
\citep{jon90,jon91,jam92}
```

into [Jones et al. 1990; 1991, James et al. 1992].

Example 2, `\bibpunct[;]{}{()}{,}{a}{}{;}` changes the output of

```
\citep[and references therein]{jon90}
```

into (Jones et al. 1990; and references therein).

Usually the citation style is determined by the journal for which one is writing, and is as much a part of the bibliography style as everything else. The `natbib` package allows punctuation definitions to be directly coupled to the `\bibliographystyle` command that must always be present when L^AT_EX is used. It is this command that selects

The natbib package

the .bst file; by adding such a coupling to natbib for every .bst file that one might want to use, it is not necessary to add \bibpunct explicitly in the document itself, unless of course one wishes to override the preset values.

Such a coupling is achieved by defining a command \bibstyle@bst, where bst stands for the name of the .bst file. For example, the American Geophysical Union (AGU) demands in its publications that citations be made with square brackets and separated by semi-colons. I have an agu.bst file to accomplish most of the formatting, but such punctuations are not included in it. Instead, natbib has the definition

```
\newcommand{\bibstyle@agu}
{\bibpunct{[ ]}{;}{a}{,}{,~}}
```

These style defining commands may contain more than just \bibpunct. Some numerical citation scheme require even more changes. For example, the journal *Nature* not only uses superscripted numbers for citations, it also prints the numbers in the list of references without the normal square brackets. To accommodate this, natbib contains the style definition

```
\newcommand{\bibstyle@nature}
{\bibpunct{}{}{,}{s}}{\textsuperscript{,}}%
\gdef\NAT@biblabelnum##1{##1.}}
```

The redefined \NAT@biblabelnum command specifies how the reference numbers are to be formatted in the list of references itself. The redefinition must be made with \gdef, not \def or \renewcommand.

The selected punctuation style and other redefinitions will not be in effect on the first L^AT_EX run, for they are stored to the auxiliary file for the subsequent run.

The user may add more such definitions of his own, to accommodate those journals and .bst files that he has. He may either add them to his local copy of natbib.sty, or better put them into a file named natbib.cfg. This file will be read in if it exists, adding any local configurations. Thus such configurations can survive future updates of the package. (This is for L^AT_EX 2_ε only.)

Any explicit call to \bibpunct has priority over the predefined citation styles.

A preprogrammed citation style is normally invoked by the command \bibliographystyle, as described above. However, it may be that one wants to apply a certain citation style to another bibliography style. This may be done with \citestyle, given *before* \begin{document}. For example, to use the plainnat bibliography style (for the list of references) with the *Nature* style of citations (superscripts),

```
\documentclass{article}
\usepackage{natbib}
. . . . .
\citestyle{nature}
\begin{document}
\bibliographystyle{plainnat}
. . . . .
```

All changes to the citation style, including punctuation, must be made before \begin{document}, which freezes the citation style.

41.4 Priority of style commands

The citation style (punctuation and mode) can be selected by means of the \bibpunct, \citestyle, and predefined \bibstyle@bst commands. They can also be selected by L^AT_EX 2_ε options (p. 35). What happens if there are several conflicting selections?

The lowest priority is assigned to the predefined \bibstyle@bst commands, since they are implicit and not obvious to the user. The L^AT_EX 2_ε options have the next priority. Finally, any selection by \bibpunct and/or \citestyle overrides those of the other methods.

41.5 Other formatting options

The list of references normally appears as a \section* or \chapter*, depending on the main class. If one wants to redesign one's own heading, say as a numbered section with \section, then \bibsection may be redefined by the user accordingly.

Redefine \bibsection to the desired sectioning command for introducing the list of references. This is normally \section* or \chapter*.

The list of references is normally printed in the same font size and style as the main body. However, it is possible to define \bibfont to be font commands that are in effect within the thebibliography environment.

Redefine `\bibfont` to be a font declaration, e.g. `\small` to apply to the list of references.

The list of references for author–year styles uses a hanging indentation format: the first line of each reference is flush left, the following lines are set with an indentation from the left margin. This indentation is 1 em by default but may be changed by redefining (with `\setlength`) the length parameter `\bibhang`.

The indentation after the first line of each reference is given by `\bibhang`; change this with the `\setlength` command.

The vertical spacing between references in the list, whether author–year or numerical, is controlled by the length `\bibsep`. If this is set to 0 pt, there is no extra line spacing between references. The default spacing depends on the font size selected in `\documentclass`, and is almost a full blank line. Change this by redefining `\bibsep` with `\setlength` command.

The vertical spacing between references is set by `\bibsep`; change this with the `\setlength` command.

41.6 Automatic Indexing of Citations

If one wishes to have the citations entered in the `.idx` indexing file, it is only necessary to issue `\citeindextrue` at any point in the document. All following `\cite` commands, of all variations, then insert the corresponding entry to that file. With `\citeindexfalse`, these entries will no longer be made.

The `\bibitem` commands in the `thebibliography` environment will also make index entries. If this is not desired, then issue `\citeindexfalse` before `\bibliography` or `\begin{thebibliography}`. Of course, `\makeindex` must also be issued in the preamble to activate indexing, as usual. Otherwise, no indexing is done at all. Make sure that the document has been processed at least twice after the last `BIBTEX` run before running the `makeindex` program.

The form of the index entries is set by the internal `\NAT@idxtxt`, which can be redefined by hackers if wanted (in the `natbib.cfg` file please). By default, it prints the short author list plus date in the current parenthesis style.

The `natbib` package can also be used with the `index` package of David M. Jones. The order in which the packages are loaded is not important.

In that package, multiple index lists may be made by means of a `\newindex` command. For example, it may be desirable to put all the citation indexing into a separate list. First that list must be initiated with, e.g.,

```
\newindex{cite}{ctx}{cnd}{List of Citations}
```

and then the automatic citation indexing associated with this list with the `natbib` command

```
\renewcommand{\citeindextype}{cite}
```

See the documentation for `index.sty` for details.

41.7 Hyper_TE_X compatibility

`natbib` is compatible with the `hyperref` package of Sebastian Rahtz and Yannis Haralambous, for use with Hyper_TE_X. The compatibility is of a mutual nature: both packages contain coding that interact with that of the other.

41.8 Multiple bibliographies in one document

`natbib` is compatible with the `chapterbib` package of Donald Arseneau and Niel Kempson, which makes it possible to have several bibliographies in one document. The usual application is to have bibliographies in each chapter of a book, especially if they have been written by different authors.

The `chapterbib` package works in a very natural way for the author; only the editor who puts all the chapters together into one book has to do some extra work.

The package makes use of the `\include` command, and it is in fact every `\included` file that has its own bibliography. For large books, it makes very good sense to take advantage of this feature in any case.

To review the use of `\include`, recall that the main file

```
\documentclass{...}
\includeonly{ch2}
\begin{document}
  \include{ch1}
  \include{ch2}
  \include{ch3}
\end{document}
```

will process only the file `ch2.tex` as though the files `ch1.tex` and `ch3.tex` were also present. That is, all counters, especially the page and section numbers, as well as cross-referencing definitions, will function as if the whole document were processed. The trick is that each `\included` file has its own `.aux` file containing these definitions,

The *natbib* package

and they are all read in every time, even if the corresponding `.tex` file is not. The `.aux` files also contain the citation information for `BIBTEX`, something that the `chapterbib` package exploits.

If `\usepackage{chapterbib}` has been given, the keys in each `\cite` and `\bibitem` command are associated with the current `\included` file and are distinguished from the identical key in a different file. Each of these files must contain its own `\bibliography` and `\bibliographystyle` commands. One processes `BIBTEX` on each file separately before processing it under `LATEX` (at least twice).

Special considerations for natbib and chapterbib

The `chapterbib` package provides an option `sectionbib` that puts the bibliography in a `\section*` instead of `\chapter*`, something that makes sense if there is a bibliography in each chapter. This option will not work when `natbib` is also loaded; instead, add the option to `natbib`. (The `sectionbib` option can always be given, but it only has meaning for the `book` and `report` classes, or for classes derived from them.)

Every `\included` file (with citations) must contain its own `\bibliography` command where the bibliography is to appear. The database files listed as arguments to this command can be different in each file, of course. However, what is not so obvious, is that each file must also contain a `\bibliographystyle` command, *preferably with the same style argument*. If different bibliography styles are specified for different files, then the preprogrammed citation style (punctuation and citation mode) will be that of the first bibliography style given. The preprogrammed citation styles can only be changed in the preamble (see p. 39), something that guarantees a uniform style for the entire document.¹²

The order in which the `chapterbib` and `natbib` packages are loaded is unimportant.

41.9 *Sorting and compressing numerical citations*

Another package by Donald Arseneau, `cite.sty`, reimplements the entire (numerical) citation system such that one can control the punctuation and citation format, all of which is done by `natbib` as well. However, it also can sort and compress numerical citations, something that is required by some journals.

What this means is that when multiple citations are given with a single `\cite` command, the normal order of the numbers is in the sequence given. This is usually a wild list of numbers, such as [4,2,8,3]. With the `cite` package, this list becomes [2–4,8].

It is impossible to make the `cite` and `natbib` packages compatible, since both reimplement `\cite` from scratch. Instead, I have taken over some of the coding from `cite.sty`, modifying it for `natbib`. This coding is activated by including one of the options `sort` or `sort&compress` in the `\usepackage` command.

For author–year citations, the option `sort` orders the citations in a single `\citep` or `\citet` command into the sequence in which they appear in the list of references. This is normally alphabetical first, year second. This should avoid citations of the type: “James et al. (1994b,a)”. For author–year mode, the `sort&compress` option is identical to `sort`.

Do not use the `cite` package with `natbib`; rather use one of the options `sort` or `sort&compress`. These also work with author–year citations, making multiple citations appear in the order they appear in the reference list.

41.10 *Long author list on first citation*

A feature that has often been requested by otherwise happy users of `natbib` is one that is found in the `harvard` package as standard: with the first citation of any reference, the full author list is printed, and afterwards only the abbreviated list. One can control this with `\citet*` for the first citation, and `\citet` or `\citep` thereafter. However, the automatic feature is very desirable. This can be activated with the option `longlistfirst`.

Long author list on first citation

Use option `longlistfirst` to have first citation automatically give the full list of authors. Suppress this for certain citations with `\shortcites{key-list}`, given before the first citation.

Some references have so many authors that you want to suppress the automatic long list only for them. In this case, issue

```
\shortcites{key-list}
```

before the first citations, and those included in *key-list* will have a short list on their first citation.

Full author lists can still be forced at any time with the starred variants.

¹²It would be relatively easy to allow changes in style anywhere in the document, but this strikes me as bad policy. However, it is provided for with the `docstrip` option `nopreonly`.

42 Numerical citations with author–year styles

In earlier versions, each `.bst` file could be used either for numerical or author–year citation mode, without any possibility of switching. Obviously numerical `.bst` files can never be used for author–year citations (the information is missing in the auxiliary file) but there is no reason why the converse should not work.

I have been frequently asked if I could implement this feature, and I at first replied that it would be quite easy. That was a mistake. The numerical information was missing in the auxiliary files, so the `\cite` commands could never access it. I have now recoded `natbib` so that the separation of author and year information comes before the information is written to the auxiliary file, and furthermore, a sequential number is also written. This permits numerical citations with *any* of the author–year bibliographic style files.

42.1 *Selecting numerical mode*

By default, `natbib` is in author–year mode. This can be changed by

1. selecting a numerical bibliography style with predefined citation style, defined either in the package or in the local configuration file;
2. giving options `numbers` or `super` to the `\usepackage` command;
3. issuing `\bibpunct` with the 4th mandatory argument set to `n` or `s`;
4. issuing `\citestyle` with the name of a predefined numerical bibliography style.

The methods are listed in order of increasing priority.

The `natbib` package will automatically switch to numerical mode if any one of the `\bibitem` entries fails to conform to the possible author–year formats. There is no way to override this, since such an entry would cause trouble in the author–year mode.

There are certain special ‘numerical’ styles, like that of the standard `alpha.bst`, which include a non-numerical label in place of the number, in the form

```
\bibitem[ABC95]{able95}
```

As far as `natbib` is concerned, this label does not conform to the author–year possibilities and is therefore considered to be numerical. The citation mode switches to numerical, and `\cite{able95}` prints `[ABC95]`.

42.2 *New bibliography styles*

I provide three new `.bst` files to replace the standard \LaTeX numerical ones (`plainnat.bst`, `abbrvnat.bst` and `unsrtnat.bst`). These produce reference lists in the same style as the corresponding standard `.bst` file, but can only be used with `natbib`. The advantage is that they can be used in both numerical and author–year mode.

In fact, `unsrtnat.bst` is rather silly for author–year citations: the references are listed in the order they were cited, which makes it difficult to find them. It is only included in case an author wishes to switch from `plainnat` (author–year) to the unsorted numerical mode, retaining the `\citeauthor` and `\citeyear` commands in his paper.

It is not possible to produce a modified version of `alpha.bst` since it would conflict with the `\bibitem` format of `natbib`.

43 Local configuration

For $\LaTeX 2_{\epsilon}$, it is possible to add a local configuration file `natbib.cfg`, which is read in, if it exists, at the end of the package. It may thus contain coding to supcede that in the package, although its main purpose is to allow the user to add his own `\bibstyle@bst` definitions to couple citation punctuation with local bibliography styles.

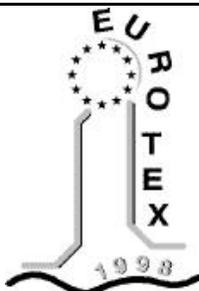
44 \LaTeX freelance editors sought

The journals department of Taylor and Francis use a number of freelance copy-editors and due to a recent increase in disk submission are recruiting freelancers with on-screen facilities, in particular those able to work with \LaTeX files.

Contact:

Julie Knight
Journal Production Administrator
Taylor & Francis Ltd.
1 Gunpowder Square
London EC4A 3DE
0171 583 0490 ext: 547
julie.knight@tandf.co.uk

45 EuroT_EX'98 and TUG 98



46 EuroT_EX'98: Saint-Malo, France — March 29th to 31st, 1998

GUTenberg is organizing the 10th European T_EX Conference, which will be a part of WEPT'98, the second international Week on Electronic Publishing and Typography held in Saint-Malo (Brittany, France) from March 29th to April 3rd, 1998.

EuroT_EX'98 will be the meeting-place for the leading actors in the field of scientific and technical publishing, especially those using T_EX as their typesetting engine.

The conference starts on the morning of Sunday March 29th and lasts until midday Wednesday April 1st, 1998. On Sunday four tutorials will be given, two in English and two in French; they will be open to all, including beginners. The main conference, on Monday and Tuesday, will feature many international experts on document publishing. Finally, on Wednesday morning April 1st, a choice will be offered between two tours, one to the medieval township of Mont Saint-Michel and the other to its Library of ancient manuscripts and early printed books in Avranches.

EuroT_EX'98 will also celebrate the 10th anniversary of the foundation of *GUTenberg*, with a buffet dinner served on Monday evening.

46.1 Connection with WEPT

EuroT_EX'98 is part of WEPT'98, the second “Week on Electronic Documents” with related conferences.

These include:

- **PODDP'98**: Workshop on Principles of Digital Document Processing, March 29th and 30th, 1998. (mirror site in USA)
- **EuroT_EX'98**
- **RIDT'98**: Raster Imaging and Digital Typography 98, March 30th to April 1st, 1998.
- **EP'98**: Electronic Publishing 98, April 1st to 3rd, 1998.

These are scheduled in parallel, and are open to participants in other WEPT conferences.

46.2 Calendar

(these dates may be modified later by the organizing committee)

- January 1998 : Registrations are open.
- February 15th, 1998 : Deadline for registration at preferential rate.
- February 15th, 1998 : Deadline for hotel booking
- March 6th, 1998 : Final deadline for EuroT_EX'98 registration.
- March 20th, 1998 : Deadline for refundable cancellation.
- March 28th, 1998 : Welcome.
- March 29th, 1998 : EuroT_EX'98 tutorials .
- March 30th–31st, 1998 : EuroT_EX'98 conference .
- April 1st, 1998 : Mont Saint-Michel visit
Avranches medieval Library visit

From the beginning of January to February 15th, a special reduced registration fee is being offered.

EuroT_EX'98 and TUG 98

Registration for the conference and/or tutorials may be made directly from this Web site, or with the available registration form. Official registration forms which will be mailed around mid-January. Other forms are available on request.

46.3 Enquiries

General enquiries should be sent in English or in French to the EuroT_EX'98 office

GUTenberg BP 10

F-93220 GAGNY Principal, France

tel.: +33 (0)1 30 87 06 25

fax: +33 (0)1 30 87 06 25

EuroTeX98@ens.fr

47 TUG'98: The 1998 TeX Users Group Conference

The 1998 TeX Users Group Conference will take place at the Nicholas Copernicus University, Toruń, Poland from 17th to 21st August 1998. This conference will be organised and hosted by the Polish TeX Users Group, GUST.

Proposals for papers are now being solicited: preference will be given to papers which deal most directly with the theme of the conference ('Integrating TeX with the surrounding world'), but papers on related topics (e.g. DSSSL, HTML, PDF, SGML, XML, etc.) are not excluded. Proposals (which should ideally be written in English, but which may be written in another language by prior arrangement) should be sent to the TUG'98 Programme Committee (email tug-98-papers@mail.tug.org).

Each proposal should include the title, name, address, e-mail address and affiliation (where appropriate) of the proposer, together with an extended abstract (the latter should correspond to approximately one side of A4. In addition, there should be an estimate of the necessary time for verbal presentation (excluding questions: a further five minutes will be allowed for questions after each talk), and a further estimate of the number of printed pages which will be required to reproduce the full text of the article using the [L]TUGproc macros. Any special needs should also be stated (for example, colour pages in the preprints/proceedings; e-TeX/pdfTeX/Omega/etc. to process the paper, ...).

48 Bursary for UKTUG members to attend TeX conferences

In addition to the general Bursary funds, the Committee of UKTUG has set aside the sum of \$500 to support one or more members of UKTUG in attending overseas TeX-related conferences. There are two main conferences this year which could be supported from this fund, EuroTeX'98 in St Malo (France) at the end of March, and TUG'98 in Toruń; (Poland) in mid August. If you want assistance from this fund, whether for EuroTeX or TUG98, please contact the Secretary as soon as possible (see contact details on back page).

49 Donald Knuth's 60th birthday

Jonathan Fine

Recently I noticed from the \TeX book that Knuth would be 60 in 1998. I thought it might be nice to organise a meeting to express our appreciation. And so I wrote to him, suggesting this. I also took the opportunity to ask him about his relationship and attitude to the NTS project. His reply follows. (Photocopies were circulated to those present at the 1998 AGM.)

Dear Jonathan,

Thanks for your letter.

About birthdays, I believe (with the Beatles) that a 64th birthday is vastly more interesting than a 60th birthday. Thus I will be looking forward most eagerly to January 10, 2002.

However, I am of course delighted that you and others might still appreciate some of my work after having ben exposed to it for so long. Thanks for the sentiment.

About NTS and Ω , etc., I have always encouraged work that leads to improvements in typography. I want \TeX itself to remain a fixed point forever, warts and all, entirely my responsibility; but I also believe that anything of nontrivial complexity is inherently non-optimum. So I look forward to continued improvements by conscientious people who are dedicated to quality software. (I do warn them that the task of reaching a decent fixed point is not easy!)

Best wishes for a successful AGM. I received my invitation two days ago and it looks like a splendid meeting has been planned, although of course I cannot take the time to travel while working on TAOCP.

Sincerely,

Donald E. Knuth

It is at present too early, in my opinion, to make plans for Knuth's 64th (octal 100 and binary 1,000,000) birthday. But in a year, or perhaps two, we should allow the prospect of such a meeting, to celebrate his contribution, to come forward in our minds.

50 Provisional minutes of the UKTUG AGM, 26 November 1997

David Carlisle

Present: 19 members, including Robin Fairbairns in Chair. David Rhead had sent apologies for absence.

51 1. Report of the 1996 AGM

After an initial welcoming address, Robin Fairbairns, as Chair, asked the meeting to accept the minutes of the 1996 AGM which had been distributed to attendees. Peter Abbott made one correction, the date by which membership fees must be paid to receive a 10% discount should be January 31 not December. With this one correction, the minutes were accepted by the meeting.

52 2. Chair's Report

Printed copies of the Chairman's report were distributed at the meeting. Robin Fairbairns spoke substantially on the points in the printed text, especially the point urging members to volunteer for work on the committee. He pointed out that although by the letter of the constitution (copies of which had been made available to all attendees, thanks to Jonathan Fine) nominations for committee should be submitted a week before the AGM, the committee was perilously close to being inquorate (5) and so it had been agreed by the outgoing committee to be flexible and accept nominations for committee from attendees at the meeting. (This point was returned to in item 8.) One new item was the decision to support the newly formed TUG India by sending one committee member (to be decided) to its inaugural meeting in January 1998.

53 3. Treasurer's Report

A financial statement had been sent to all members with the notice of the meeting. Peter Abbott explained a few more mysterious points.

1. The item for expenses itemised as 'Chair' were in fact running costs for Baskerville (Both items being Robin Fairbairns).
2. The L^AT_EX³ funds, since the close of the UKTUG financial year had been split off into a separate account and would not therefore be shown on future UKTUG statements.
3. From a question from Malcolm Clark, Peter confirmed the group's finances were stable (growing slightly) mainly due to CD sales and extra members, and that therefore the group could afford to run meetings such as today's at a loss.

Philip Taylor noted that the Bursaries that the group had made in the past had been gratefully received, and he thanked the group on behalf of the recipients and the organisers of the conferences supported.

54 4. Appointment of Auditors

Peter Abbott reported that a change in the law meant that it was not necessary for a the group accounts to be 'audited' but rather just 'inspected by a competent independent person'. Peter hoped to find such a person within a few weeks.

55 5. Setting annual membership fee

"The membership fee for 1998 shall be £20.00 for full membership, or £5.00 for full-time student membership."

Peter Abbott again pointed out that the date for discount should be January 31 rather than December. (This mistake having been copied over to the present agenda from last year's minutes.) After some clarification that these fees were identical to last year, they were accepted by the meeting without vote.

56 6. Setting annual institutional membership fee

“The membership fee for 1998 shall be £100.00 for institutional membership.”

Similarly the institutional membership fees were accepted without vote to be the same as last year. Peter announced he had reached agreement with TUG that institutions should in future be offered joint TUG/UKTUG membership.

At this point Jonathan Fine proposed a procedural motion to change the order of the Agenda. (He had circulated an alternative agenda to attendees.) The motion was not seconded, and so discussion moved to the next point on the original Agenda.

57 7. Confirmation of Philip Taylor as Chair of the Group

Robin Fairbairns explained that Baskerville was late due to his illness and finally due to technical failures with the printer used to produce the issue at Cambridge. This resulted in the fact that the announcement of an election for Chair slipped and so the committee agreed to announce the nominations should close on 24th October. Nomination of Jonathan Fine was received (26th September) before the announcement went out, and shortly afterwards from Philip Taylor. Shortly before close of Nominations, Jonathan withdrew his nomination, and so Philip was elected to the Chair, unopposed.

At this point, as is traditional, Robin vacated the Chair, and Philip Taylor took over the meeting.

58 8. Election of Committee Members

Of the current committee:

- Peter Abbott (Treasurer) continues to stand.
- Sebastian Rahtz continues.
- Robin Fairbairns was standing down as Chair, but had been nominated for the new Committee
- David Carlisle was standing down (3 year rule);
- Mark Wooding (co-opted shortly after 96 AGM) was standing down (as co-opted) but was nominated on the day to stand in the next committee;
- Malcolm Clark standing down (2/3 rule) but was nominated on the day to stand in the next committee;
- Kaveh Bazargan standing down (2/3 rule) but was nominated on the day to stand in the next committee.

Nominations had been received from Jonathan Fine and Dominik Wujastyk, and The following members were also nominated on the day (according to the ruling announced in the Chairman’s statement): Kim Roberts, David Hardy and Roy Everett.

As fewer than 12 places were filled, all people nominated were elected without vote.

59 9. Report on Baskerville

60 10. T_EXnical services to members

A printed report on Baskerville and T_EXnical services to members had been distributed at the meeting. Sebastian Rahtz spoke briefly introducing the items. They were accepted with no substantive comment.

61 11. Motions proposed by members

Jonathan Fine had proposed four motions to the meeting, they were taken in turn, and voted on by paper ballot. The papers were in fact taken in and counted at the end of all the motions, but these minutes record the results by each motion. There were 19 attendees at the meeting, and one proxy vote. Dominik Wujastyk volunteered to count the votes.

61.1 Motion 11a)

“This meeting hopes all members of UK TUG will be able to resolve or put to one side their personal differences, so that they can better cooperate for the common good.”

Jonathan spoke for the motion, and in particular suggested that a meeting to celebrate Knuth’s 64th birthday would help heal the divisions in the community. After some discussion as to whether this was a suitable subject for a motion, and a failed procedural motion put to leave this on the table, it was put to a vote and carried: 8 for and 4 against.

61.2 *Motion 11b)*

“This meeting asks the Committee to produce a printed list of members of UK TUG, to be distributed to the members, for their personal use. Any member may, if they so wish, not appear in the list.”

The discussion centered mainly on the data protection act, but it seemed that it would be allowable to ask if members wanted to be added to such a mailing list, although possibly not to directly use the existing membership records for this purpose. However the motion failed: 8 for and 10 against.

61.3 *Motion 11c)*

“This meeting asks the Committee to allow those persons who through UK TUG pay dues to TUG to indicate that they wish these dues to be held in trust, until the Committee of UK TUG can be satisfied that TUG’s financial affairs are in proper order.”

Jonathan withdrew this motion for lack of time.

61.4 *Motion 11d)*

“This meeting resolves that the position of Chair shall remain empty as from this AGM, until it can be filled by an election, to take place in the first quarter of 1998.”

Jonathan outlined his objections to the conduct of the last two elections. He was concerned that their conduct was probably unconstitutional, and believed he had been discriminated against. Robin Fairbairns referred the meeting to the points made under item 7.

The motion failed: 2 for and 18 against.

62 12. The Future of UKTUG Meetings, and possible topics

Philip Taylor asked for suggestions on the number of meetings per year the group should hold, and for possible topics. Various numbers were suggested with a consensus perhaps of something in the range 2–3. Suggested topics were: ‘ \TeX for interactive Documents’ (otherwise known as Con \TeX t for all); ‘ \TeX in Publishing’ and ‘ \TeX in Education’

There was discussion (principally involving Malcolm Clark and Sebastian Rahtz) on whether UKTUG should consider hosting a future larger meeting, perhaps Euro \TeX , or something tied to Knuth’s 64th Birthday.

63 13. Any other business

No items raised.

64 UKTUG Information

65 The 1997–98 UKTUG committee

Chair: Philip Taylor: p.taylor@vms.rhnc.ac.uk

Treasurer & Membership Secretary:

Peter Abbott: peter.abbott@cl.cam.ac.uk

Secretary: Jonathan Fine: j.fine@pmms.cam.ac.uk

Kaveh Bazargan: kaveh@focal.demon.co.uk

Malcolm Clark: malcolm.clark@kcl.ac.uk

Roy Everett: purpose@compuserve.com

Robin Fairbairns: Robin.Fairbairns@cl.cam.ac.uk

David Hardy: editent@btinternet.com

Sebastian Rahtz: s.rahtz@elsevier.co.uk

Kim Roberts: robertsk@oup.co.uk

Mark Wooding: mdw@ebi.ac.uk

Dominik Wujastyk: ucgadkw@ucl.ac.uk

66 Contacting UKTUG

Please send UKTUG subscriptions, and book or software orders, to the Treasurer: Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Fax/telephone: 0121 476 2159.

General enquires should be sent to the Secretary: Jonathan Fine, 203 Coldhams Lane, Cambridge CB1 3HY. Telephone: 01223 215389.

UKTUG maintains pages on the World Wide Web at <http://www.tex.ac.uk/UKTUG/> Email enquiries about UKTUG to uktug-enquiries@tex.ac.uk.

67 Baskerville

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

68 Book Discounts for UKTUG 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 20%) price, is distributed occasionally with *Baskerville*, but is always available from the Treasurer.

Please add £1.50 for the first book and 50p for each book after the first on the same order, for despatch to a single address

We are only allowed to offer this service to **current** members of the UKTUG and/or members of TUG. Please send your order and cheque (in UK £) to Peter Abbott. Make cheques payable to 'UKTUG' please. All books will be routed through UKTUG. *In all cases* please notify Peter Abbott by email, phone, fax or letter when books are delivered. This means that provided the book(s) are in stock, it will normally take at least a week from receipt of order to delivery of the book(s).

69 Obtaining T_EX from CTAN

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.

For interfaces and catalogues of CTAN, you are referred to Graham Williams' *T_EX and L^AT_EX Catalogue* which reprinted from *Baskerville*