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.
I Editorial

This is the first edition of Baskerville entirely devoted to a single topic. It arose from discussion within your committee of what we might reasonably do which helps our members, but which isn’t already done elsewhere. We hope it will prove useful to you. We would welcome comments on the utility or otherwise of the article, and on ways it could be improved; letters to the editor are always welcome. Future uses of this edition could include inserting it into a ‘new members pack’, publishing updated questions, and possibly republishing the whole thing.

This edition of Baskerville was processed using a testing copy of the December 1994 release of \LaTeX\, but none of the answers to questions assume that that version is available (it’s scheduled for public release in the middle of December). This edition was printed in Cambridge on a Xerox Docutech, in Monotype Baskerville.
II The New T\TeX\ FAQ: Your 100 Questions Answered

The UK T\TeX\ Users Group Committee
after the original maintained by
Bobby Bodenheimer

Contents

A Introduction ........................................ 5
B The Background ....................................... 6
1 What is \TeX? ........................................ 6
2 How should I pronounce “\TeX”? .................... 6
3 What is METAFONT? .................................. 6
4 What is \LaTeX? ...................................... 6
5 How should I pronounce “\LaTeX”? “\LaTeX{}”? ........ 7
6 Should I use \plain or \LaTeX{}? .................... 7
7 What are the AMS packages (\ams,\LaTeX, etc.)? .... 7
8 What is \ams\LaTeX{}? ................................ 7
9 What is \breqn ........................................ 7
10 What is Lollipop? ................................... 7
11 What is Texpref? ................................... 7
12 If \TeX\ is so good, how come it’s free? .......... 8
13 What is the future of \TeX{}? ....................... 8
14 What are TUG and TUGboat? ....................... 8
15 Are there nationally-based user groups, too? .... 8
16 TUG Technical Working Groups .................. 9
C Documentation and Help ............................ 10
17 Books on \TeX\ and its relations .................. 10
18 Where to find this article ......................... 11
19 Mailing lists about \TeX\ and its friends .......... 11
20 Bt\TeX\ Documentation ............................. 11
21 The \pictex\ manual ................................ 11
22 Finding \LaTeX{}/\breqn macro packages ......... 11
23 Finding \LaTeX{}\LaTeX{}-related packages ......... 12
D Bits and pieces of \TeX\ ............................ 12
24 Virtual fonts ..................................... 12
25 ‘special’ commands ................................. 12
26 Documented \LaTeX{} sources (.dtx files) ......... 12
E Acquiring the Software ............................. 13
27 Repositories of \TeX\ materials ................... 13
28 Contributing a file to the archives ............. 13
29 Finding new fonts .................................. 13
30 \TeX\ CD-ROMs .................................... 14
F \TeX\ Systems ....................................... 14
31 \LaTeX{}/\breqn for different machines .......... 14
32 \TeX\-friendly editors and shells ................ 15
33 Commercial \TeX\ implementations ............... 15

reprinted from Baskerville  Volume 4, Number  6
A Introduction

This article was prepared by the Committee of the UK \TeX Users Group (UK TUG)\footnote{For 1993-94: Peter Abbott, Rosemary Bailey, Malcolm Clark, Robin Fairbairns, Jonathan Fine, Carol Hewlett, Alan Jeffrey, Sebastian Rahtz, and Chris Rowley; in addition, for 1994–95, David Carlisle} as a development of a regular posting to the Usenet newsgroup comp.text.tex that has been maintained for some time by Bobby Bodenheimer (bobby@hot.caltech.edu).

Usenet is a mechanism for exchanging articles between people who share interests or need\footnote{As its name implies, Usenet is a means of using some sort of network; in the earliest days the network was made by stringing together a series of telephone lines, but nowadays Usenet is most often carried over the Internet}, a newsgroup is an area within Usenet carrying a particular class of articles. Since a common sort of article asks for help, advice or information, and since certain of these questions are regularly repeated (often with monotonous regularity), some public-spirited souls took to writing articles which listed “Frequently Asked Questions” and answers to them. Many members of UK TUG do not have access to Usenet, but could be expected to value the answers about \TeX that have accumulated over the years; so we decided to update the list and publish it in Baskerville; we are grateful to Bobby for his permission to use his article in this way. As a quid pro quo, we are making the source of the article freely available, and it can be compiled by anyone who runs a production \TeX\ ensuring the availability, and it can be compiled by anyone who runs a production \TeX\ and has the required fonts. It is the committee’s hope that it will also be possible for the content of this article to feed back to the world-wide \TeX community via Bobby’s regular posting.

In addition, a translation of the article is available on the World-Wide Web, via URL http://www.cogs.susx.ac.uk/cgi-bin/texfaq2html?introduction=yes\footnote{This is a temporary URL; a final home for the document is to be provided in due course}.

We have rearranged Bobby’s article quite a lot, and have added new questions and answers on the basis of our experience of answering questions about \TeX, writing documents in \TeX, and developing macros for \TeX, over the years. We have also pruned it to take account of the changes that have happened in the world of \TeX since Bobby first started.

The committee is grateful for help and advice, from the following outside its number: Barbara Beeton, Karl Berry, Damian Cugley, Michael Downes, Berthold Horn, Werner Icking, Ted Nieland, Pat Rau, Joachim Schrod, Philip Taylor, Ulrik Vieth, Rick Zaccone and Reinhard Zierke.

---

---

1 For 1993-94: Peter Abbott, Rosemary Bailey, Malcolm Clark, Robin Fairbairns, Jonathan Fine, Carol Hewlett, Alan Jeffrey, Sebastian Rahtz, and Chris Rowley; in addition, for 1994–95, David Carlisle

2 Usenet, as its name implies, is a means of using some sort of network; in the earliest days the network was made by stringing together a series of telephone lines, but nowadays Usenet is most often carried over the Internet

3 This is a temporary URL; a final home for the document is to be provided in due course
Finding the Files

Unless otherwise specified, all files mentioned in this article are available from a CTAN archive, or from one of their mirrors. Question 27 gives details of the CTAN archives, and how to retrieve files from them. If you don’t have access to the internet, question 30 tells you of sources of CD-ROMs that offer snapshots of the archives.

The reader should also note that the first directory name of the path name of every file on CTAN has been elided from what follows, for the simple reason that it’s always the same (\texttt{tex-archive/}).

To avoid confusion, we’ve elided the full stop \(^\text{4}\) from the end of any sentence whose last item is a path name (note that such sentences only occur at the end of paragraphs). Though the path names are set in a different font from running text, it’s not easy to distinguish the font of a single dot!

B The Background

1 What is \LaTeX?\footnote{‘Full stop’ (British English) = ‘period’ (American English)}

\LaTeX{} is a typesetting system written by Donald E. Knuth, who says in the Preface to his book on \LaTeX{} (see question 17) that it is “intended for the creation of beautiful books—and especially for books that contain a lot of mathematics”.

Knuth developed a system of ‘literate programming’ to write \LaTeX{}, and he provides the literate (WEB) source of \TeX{} free of charge, together with tools for processing the WEB source into something that can be compiled and something that can be printed; there’s never any mystery about what \TeX{} does. Furthermore, the WEB system provides mechanisms to port \TeX{} to new operating systems and computers; in order that one may have some confidence in the ports, Knuth supplied a test by means of which one may judge the fidelity of a \TeX{} system. \TeX{} and its documents are therefore highly portable.

\TeX{} is a macro processor, and offers its users a powerful programming capability. For this reason, \TeX{} on its own is a pretty difficult beast to deal with, so Knuth provided a package of macros for use with \TeX{} called \texttt{plain \TeX}; \texttt{plain \TeX} is effectively the minimum set of macros one can usefully employ with \TeX{}, together with some demonstration versions of higher-level commands (the latter are better regarded as models than used as-is). When people say they’re “programming in \TeX{}”, they usually mean they’re programming in \texttt{plain \TeX}.

2 How should I pronounce “\TeX{}”?\footnote{‘Full stop’ (British English) = ‘period’ (American English)}

The ‘X’ stands for the Greek letter Chi (χ), and is pronounced by English-speakers either a bit like the ‘ch’ in ‘loch’ ([x] in the IPA) or like ‘k’. It definitely is not pronounced ‘ks’.

3 What is \METAFONT?\footnote{‘Full stop’ (British English) = ‘period’ (American English)}

\METAFONT{} was written by Knuth as a companion to \TeX{}; whereas \TeX{} defines the layout of glyphs on a page, \METAFONT{} defines the shapes of the glyphs and the relations between them. \METAFONT{} details the sizes of glyphs, for \TeX{}’s benefit, and details the rasters used to represent the glyphs, for the benefit of programs that will produce printed output as post processes after a run of \TeX{}.

\METAFONT{}’s language for defining fonts permits the expression of several classes of things: first (of course), the simple geometry of the glyphs; second, the properties of the print engine for which the output is intended; and third, ‘meta’-information which can distinguish different design sizes of the same font, or the difference between two fonts that belong to the same (or related) families.

Knuth (and others) have designed a fair range of fonts using \METAFONT{}, but font design using \METAFONT{} is much more of a minority skill than is \TeX{} macro-writing. The complete \TeX{}-user nevertheless needs to be aware of \METAFONT{}, and to be able to run \METAFONT{} to generate personal copies of new fonts.

4 What is \LATEX{}?\footnote{‘Full stop’ (British English) = ‘period’ (American English)}

\LATEX{} is a \TeX{} macro package, written by Leslie Lamport, that provides a document processing system. \LATEX{} allows markup to describe the structure of a document, so that the user need not think about presentation. By using document classes and add-on packages, the same document can be produced in a variety of different layouts.

Lamport says that \LATEX{} “represents a balance between functionality and ease of use”. This shows itself as a continual conflict that leads to the need for such as the present article: \LATEX{} can meet most user requirements, but finding out how is often tricky.
5 How should I pronounce “\LaTeX”? “\LaTeX 2\epsilon”? Lamport never recommended how one should pronounce \LaTeX, but a lot of people pronounce it ‘Lay \TeX’ or perhaps ‘Lah \TeX’ (with \TeX pronounced as the program itself; see question 2). The ‘epsilon’ in ‘\LaTeX 2\epsilon’ is supposed to be suggestive of a small improvement over the old \LaTeX 2.09. Nevertheless, most people pronounce the name as ‘\LaTeX-two-ee’.

6 Should I use plain \TeX or \LaTeX? There’s no straightforward answer to this question. Many people swear by plain \TeX, and produce highly respectable documents using it (Knuth is an example of this, of course). But equally, many people are happy to let someone else take the design decisions for them, accepting a small loss of flexibility in exchange for a saving of brain power.

The arguments around this topic can provoke huge amounts of noise and heat, without offering much by way of light; your best bet is to find out what those around you are using, and to go with the crowd. Later on, you can always switch your allegiance; don’t bother about it.

If you are preparing a manuscript for a publisher or journal, ask them what markup they want before you develop your own; many big publishers have developed their own \LaTeX styles for journals and books, and insist that authors stick closely to their markup.

7 What are the AMS packages (\AMS\TeX, etc.)? \AMS\TeX is a \TeX macro package, originally written by Michael Spivak for the American Mathematical Society (AMS) during 1983–1985. It is described in The Joy of \TeX by Michael D. Spivak (second edition, AMS, 1990, ISBN 0-821-82997-1). It is based on plain \TeX, but provides many features for producing more professional-looking maths formulas with less burden on authors. It pays attention to the finer details of sizing and positioning that mathematical publishers care about. The aspects covered include multi-line displayed equations, equation numbering, ellipsis dots, matrices, double accents, multi-line subscripts, syntax checking (faster processing on initial error-checking \TeX runs), and other things.

As \LaTeX increased in popularity, authors asked to submit papers to the AMS in \LaTeX, and so the AMS developed \AMS-L\TeX, which is a collection of \LaTeX packages and classes that offer authors most of the functionality of \AMS\TeX.

8 What is BM\AMS-\TeX? BM\AMS-\TeX was Michael Spivak’s fusion of \AMS\TeX and \LaTeX. Its strong points are:
   • a commutative diagram package that produces very pleasing output;
   • a separate program dvipaste for producing complex tables separately from the main document (thus avoiding problems from exceeding \TeX’s main memory capacity); and
   • extensive control at the user level over the formatting of automatically-generated numbers.

However, \AMS-L\TeX had come out by the time BM\AMS-\TeX was released, so that BM\AMS-\TeX never really caught on.

9 What is Eplain? The Eplain macro package expands on and extends the definitions in plain \TeX. Eplain is not intended to provide “generic typesetting capabilities”, as do \LaTeX or Texinfo (see question 11). Instead, it provides definitions that are intended to be useful regardless of the high-level commands that you use when you actually prepare your manuscript.

For example, Eplain does not have a command \texttt{\section}, which would format section headings in an “appropriate” way, as \LaTeX’s \texttt{\section}. The philosophy of Eplain is that some people will always need or want to go beyond the macro designer’s idea of “appropriate”. Such canned macros are fine — as long as you are willing to accept the resulting output. If you don’t like the results, or if you are trying to match a different format, you are out of luck.

On the other hand, almost everyone would like capabilities such as cross-referencing by labels, so that you don’t have to put actual page numbers in the manuscript. Karl Berry, the author of Eplain, says he is not aware of any generally available macro packages that do not force their typographic style on an author, and yet provide such capabilities.

10 What is Lollipop? Lollipop is a macro package written by Victor Eijkhout; it was used in the production of his book “\TeX by Topic” (see question 17). The manual says of it:

Lollipop is \TeX made easy. Lollipop is a macro package that functions as a toolbox for writing \TeX macros. It was my intention to make macro writing so easy that implementing a fully new layout in \TeX would become a matter of less than an hour for an average document, and that it would be a task that could be accomplished by someone with only a very basic training in \TeX programming.

Lollipop is an attempt to make structured text formatting available for environments where previously only
WYSIWYG packages could be used because adapting the layout is so much more easy with them than with traditional \TeX macro packages.

The manual goes on to talk of ambitions to “capture some of the \LaTeX market share”: it’s a very witty package, but little sign of it taking over from \LaTeX is detectable... An article about Lollipop appeared in TUGboat 13(3), 1992.

11 What is Texinfo?

Texinfo is a documentation system that uses one source file to produce both on-line information and printed output. So instead of writing two different documents, one for the on-line help and the other for a typeset manual, you need write only one document source file. When the work is revised, you need only revise one document. You can read the on-line information, known as an “Info file”, with an Info documentation-reading program. By convention, Texinfo source file names end with a .texi or .texinfo extension. You can write and format Texinfo files into Info files within GNU emacs, and read them using the emacs Info reader. If you do not have emacs, you can format Texinfo files into Info files using makeinfo and read them using info.

A set of \TeX macros for formatting Texinfo files are available in macros/texinfo/texinfo.tex

12 If \TeX is so good, how come it’s free?

It’s free because Knuth chose to make it so. He is nevertheless apparently happy that others should earn money by selling \TeX-based services and products. While several valuable \TeX-related tools and packages are offered subject to restrictions imposed by the GNU General Public Licence (“Copyleft”), \TeX itself is not subject to Copyleft.

There are commercial versions of \TeX available; for some users, it’s reassuring to have paid support. What is more, some of the commercial implementations have features that are not available in free versions. (The reverse is also true: some free implementations have features not available commercially).

Usually, this article does not describe commercial versions. Question 33 lists the major vendors.

13 What is the future of \TeX?

Knuth has declared that he will do no further development of \TeX; he will continue to fix any bugs that are reported to him (though bugs are rare). This decision was made soon after \TeX version 3.0 was released; at each bug-fix release the version number acquires one more digit, so that it tends to the limit \pi. Knuth wants \TeX to be frozen at version \pi when he dies; thereafter, no further changes may be made to Knuth’s source. (A similar rule is applied to METAFONT; its version number tends to the limit e.)

There are projects (some of them long-term projects: see, for example, questions 96 and 97) to build substantial new macro packages based on \TeX. For the even longer term, there are various projects to build a successor to \TeX; see questions 98 and 99.

14 What are TUG and TUGboat?

TUG is the \TeX Users Group. TUGboat is TUG’s main journal, containing useful articles about \TeX and METAFONT. TUG also produces a newsletter for members (\TeX and TUG News), organises a yearly conference, runs training courses, sells almost all \TeX-related books, and distributes \TeX-related microcomputer software on disk. TUG has a Technical Council to coordinate \TeX-related developments (see question 16). Enquiries should be directed to:

\TeX Users Group
P. O. Box 869
Santa Barbara, CA 93102-0869
USA
Tel: (+1) 805-963-1338
Fax: (+1) 805-963-8358
Email: tug@tug.org
CTAN details: usergrps/tug

15 Are there nationally-based user groups, too?

The following groups publish their membership (etc.) information electronically on CTAN archives:

DANTE, Deutschsprachige Anwendervereinigung
\TeX e.V.
Postfach 10 18 40
D-69008 Heidelberg
Germany
Tel: (+49) 06221/2 97 66
A listing of all known groups is available as usergrps/info/usergrps.tex

16 TUG Technical Working Groups

TUG (see question 14) has an autonomous Technical Council which oversees a number of working groups on areas of common interest to the \TeX{} community. The Council has three members (current chair is Michael Ferguson, assisted by Yannis Haralambous and Sebastian Rahtz), who liaise with chair people of each working group. Each group establishes its own working methods and membership, and anyone interested in taking part should contact the chair. Suggestions for new groups should be addressed to Michael Ferguson (mike@inrs-telecom.uquebec.ca).

A brief list of the active groups follows:

WG-92-00 (IRP-TWG) — \textit{Independent Research Project TWG}.
To recognise and report to the \TeX{} Board and the \TeX{} Community on important projects which are independent of TUG but are of concern to the entire \TeX{} Community.
Contact: Alan Hoenig (ajhjj@cunyvm.cuny.edu)

WG-92-01 — \TeX{} Extended Mathematic Font Encoding.
To create font encoding standards for Mathematical fonts used in \TeX{} systems.
Contact: Barbara Beeton (bnb@math.ams.org)

WG-92-03 — \textit{Multiple Language Coordination}.
The primary purpose of this working group is to obtain, for \TeX{} systems, a consistent means for implementing, accessing, and describing the fonts, ligature rules, hyphenation patterns and other special requirements for a given linguistic group.
Contact: Yannis Haralambous (Yannis.Haralambous@univ-lille1.fr)

WG-92-04 — \TeX{} for the Disabled.
The primary purpose of this working group is as a forum for those people interested in using and/or enhancing \TeX{} to serve the needs of those with visual and other disabilities.
Contact: T.V. Raman (raman@crl.dec.com)

WG-92-05 — \TeX{} Archive Guidelines.
The purpose of this Technical Working Group is to develop guidelines for the effective management and utilisation of major \TeX{} archives, and to initiate communication among the maintainers of the existing archives for the purpose of coordination and synchronisation.
Contact: George Greenwade (bed_gdg@shsu.edu)

WG-94-07 — \TeX{} Directory Structures.
The primary purpose of this TWG is to identify a universal directory structure for macros, fonts and other related \TeX software so that recommendations can be made to all suppliers of \TeX software.

Contact: Norm Walsh (norm@ora.com)

WG-94-08 — DVI Driver Implementation and Standardisation Issues.
The major objective shall be to study the issues in the requirements of DVI Drivers imposed by changing needs and technologies, and to make recommendations for implementation and standardisation of such drivers to enhance the uniformity of their use. Work will include, but not be limited to, the examination of the use, syntax, and semantics of \special{..} commands.

Contact: Michael Sofka (sofkam@rpi.edu)

WG-94-09 — \TeX and SGML.
The major objective is to investigate the requirements and difficulties in developing an interface technology for \TeX and SGML.

Contact: Ken Dreyhaupt (kend@springer-ny.com)

WG-94-10 — \TeX and Linguistics.
The main goal is to study and discuss the requirements for typesetting linguistics in \TeX and as a means of identifying, examining, testing, and comparing macros, fonts, style files and other aids for typesetting linguistics.

Contact: Christina Thiele (cthiele@ccs.carleton.ca)

C Documentation and Help

17 Books on \TeX and its relations

While Knuth’s book is the definitive reference for \TeX, there are other books covering \TeX:

- *A Plain \TeX Primer* by Malcolm Clark (Oxford University Press, 1993, ISBNs 0-198-53724-7 (hardback) and 0-198-53784-0 (paperback))
- *\TeX: Starting from Square One* by Michael Doob (Springer Verlag, 1993, ISBN 3-540-54644-1)

For \LaTeX, see:

- *A guide to \LaTeX* Helmut Kopka and Patrick W. Daly (Addison-Wesley, ISBN 0-201-56889-6)

Of that list, only Lamport, and Goossens, Mittelbach and Samarin cover \LaTeX 2ε, though at least some new editions are reported to be in preparation.

The list for \METAFONT is rather short:


A book covering a wide range of topics (including installation and maintenance) is:

---

18 Where to find this article
The article, on which the present one is based, is posted monthly to newsgroup comp.text.tex and cross-posted to newsgroups news.answers and comp.answers. The most recently posted copy of the article is at CTAN in info/\TeX-FAQ; it is also archived at any site that archives news.answers, such as rtfm.mit.edu (18.181.0.24), and this article is available there via anonymous ftp (in the directory pub/usenet/news.answers/tex-faq). If you have access to email, but not to ftp, send an email message containing the lines "SENDME FAQ" to fileserv@shsu.edu.

A version of the article may be browsed via the World-Wide Web, at URL http://www.cogs.susx.ac.uk/cgi-bin/texfaq2html?introduction=yes

19 Mailing lists about \TeX and its friends
There are (still) people who can use networks but can’t read Usenet news; for them, not all is lost if they can send and receive email.

The UK\TeX digest and the \TeXhax digest are both operated as mailing lists. Send a message ‘subscribe uktex’ to uktex@tex.ac.uk, or ‘subscribe texhax’ to texhax@tex.ac.uk to join them.

The (rather high volume) Usenet group comp.text.tex is gatewayed to the info-tex mailing list; subscribe by sending a message ‘subscribe info-tex <your name>’ to listserv@shsu.edu. The group is also available in digest form through the ctt-digest list; subscribe by sending a message ‘subscribe ctt-Digest <your name>’ to listserv@shsu.edu; ctt-Digest is approximately the same size as info-tex, but it comes in fewer messages, which is an advantage on some systems.

Several \TeX-related lists may be accessed via listserv@vm.urz.uni-heidelberg.de. Send a message containing the line “help” to this address.

The literate programming newsgroup (see question 41) comp.programming.literate is gatewayed to the litprog mailing list; subscribe by sending a message ‘subscribe litprog <your name>’ to listserv@shsu.edu.

20 \BibTeX Documentation
\BibTeX, a program originally designed to produce bibliographies in conjunction with \LaTeX, is explained in Section 4.3 and Appendix B of Leslie Lamport’s \LaTeX manual (see question 17). The document “\BibTeXing”, contained in the file btxdoc.tex, gives a more complete description. The \BibTeX Companion (see question 17) also has information on \BibTeX and writing \BibTeX style files.

The document “Designing \BibTeX Styles”, contained in the file btxhak.tex, explains the postfix stack-based language used to write \BibTeX styles (.bst files). The file btxbst.doc is the template for the four standard styles (plain, abbrv, alpha, unsrt). It also contains their documentation. The complete \BibTeX documentation set (including the files above) is in biblio/bibtex/distrib/doc

There is a Unix \BibTeX man page in the web2c package (see question 31). Throw away any copy you may find of an obsolete man page written in 1985 before “\BibTeXing” and “Designing \BibTeX Styles” appeared.

21 The \PictureX manual
\PictureX is a set of macros for drawing diagrams and pictures. The macros are freely available in graphics/pictex; however, the \PictureX manual itself is not free. It is available for $30 ($35 with the disk) from the \TeX Users Group (see question 14). The proceeds from the sales go to Michael Wichura, the author of \PictureX, and to TUG.

22 Finding \TeX/\LaTeX macro packages
Before you ask for a \TeX macro or \LaTeX class or package file to do something, try searching the \TeX macro index written by David M. Jones (dmjones@theory.ics.mit.edu) and available in help/Tex-index/\TeX-index

The index is an excellent reference document with plenty of cross-references, but it is somewhat out of date. Until the present work to bring the index up to date is complete, you are best advised to ignore any file location information in it; having learnt of a file that seems interesting, search a CTAN archive for it (see below for the simplest method).

To use the searching features of the CTAN archives (see question 27), establish an anonymous ftp connection to a CTAN site, and type the command ‘quote site index <term>’. The archive will provide a list of files with

---

This is a temporary URL; a final home for the document is to be provided in due course
The string `<term>` in their name\(^4\) (Some ftp clients don’t implement the command ‘quote’, in most such cases, you need simply to type ‘site index `<term>`’.)

23 Finding \LaTeX\textsuperscript{2e}-related packages

A common confusion that can arise is that no .\sty or .\cls file exists on the CTAN archives for the package or class you've been recommended to acquire. For example, there are several packages listed in The \LaTeX\textsuperscript{2e} Companion (see question\(^{17}\)) without details of their location on CTAN. The file info/companion.ctan may be of help in finding such files, but it’s not a general solution.

The canonical way of distributing \LaTeX\textsuperscript{2e} packages and classes is as .\dtx files (see question\(^{20}\)). Of course, if your site index command was for ‘package.sty’, you won’t find anything — you should simply ask for ‘package’ (the search can find things on the basis of substrings of their name).

In the case that what you’re seeking is indeed distributed this way, you should retrieve the whole directory that the .\dtx file is in, which will get you all the related files, including one or more .\ins (installation) files. Process the ‘appropriate’ .\ins file using \LaTeX\textsuperscript{2e}, and you will generate the file (or files) that you want.

D Bits and pieces of \TeX

24 Virtual fonts

Virtual fonts for \TeX were first implemented by David Fuchs in the early days of \TeX, but for most people they started when Knuth redefined the format, and wrote some support software, in 1989. Virtual fonts provide a way of telling \TeX about something more complicated than just a one-to-one character mapping. The entities you define in a virtual font look like characters to \TeX (they appear with their sizes in a font metric file), but the \dvi processor may expand them to something quite different. You can use this facility just to remap characters, to make a composite font with glyphs drawn from several sources, or to build up an effect in arbitrarily complicated ways — a virtual font may contain anything which is legal in a \dvi file. In practice, the most common use of virtual fonts is to remap PostScript fonts (see question\(^{52}\)) or to build ‘fake’ maths fonts.

It is important to realise that \TeX itself does not see virtual fonts; for every virtual font read by the \dvi driver there is a corresponding .\tfm file read by \TeX. Virtual fonts are normally created in a single ASCII .\vpl file (Virtual Property List file), which includes both sets of information. The .\vpl program is then used to create the binary .\tfm and .\vf files. The commonest way (nowadays) of generating .\vpl files is to use the .\fontinst package, which is described in detail in question\(^{52}\) fonts/utilities/qdtexvpl is another utility for creating ad-hoc virtual fonts.

25 ‘special’ commands

\TeX provides the means to express things that device drivers can do, but about which \TeX itself knows nothing. For example, \TeX itself knows nothing about how to include PostScript figures into documents, or how to set the colour of printed text; but some device drivers do.

Such things are introduced to your document by means of `\special commands; all that \TeX does with these commands is to expand their arguments and then pass the command to the \dvi file. In most cases, there are macro packages provided (often with the driver) that provide a comprehensible interface to the `\special; for example, there's little point including a figure if you leave no gap for it in your text, and changing colour proves to be a particularly fraught operation that requires real wizardry. \LaTeX\textsuperscript{2e} has standard graphics and colour packages that make file inclusion, rotation, scaling and colour via `\specials all easy.

The allowable arguments of `\special depend on the device driver you’re using. Apart from the examples above, there are `\special commands in the em\TeX drivers (e.g., dvihplj, dviscr, etc.) that will draw lines at arbitrary orientations, and commands in dvitooln03 that permit the page to be set in landscape orientation.

26 Documented \LaTeX\textsuperscript{2e} sources (.\dtx files)

\LaTeX\textsuperscript{2e}, and many support macro packages, are now written in a literate programming style (see question\(^{41}\), with source and documentation in the same file. This format, known as ‘doc’, was originated by Frank Mittelbach. The documented sources conventionally have the suffix .\dtx, and should normally be stripped of documentation before use with \LaTeX\textsuperscript{2e}. Alternatively you can run \LaTeX\textsuperscript{2e} on a .\dtx file to produce a nicely formatted version of the documented code. An installation script (with suffix .\ins) is usually provided, which needs the standard \LaTeX\textsuperscript{2e} docstrip package (among other things, the installation process strips all the comments that make up the documentation for speed when loading the file into a running \LaTeX\textsuperscript{2e} system). Several packages can be included in one .\dtx file, with conditional

\(^{4}\)In fact, `<term>` is an Unix regular expression; the command is searching an index file using the command `grep`
sections, and there facilities for indices of macros etc. Anyone can write .dtx files; the format is explained in The \LaTeX Companion (see question \ref{sec:BibTeX}). There are no programs yet to assist in composition.

.dtx files are not used by \LaTeX after they have been processed to produce .sty or .cls (or whatever) files. They need not be kept with the working system; however, for many packages the .dtx file is the primary source of documentation, so you may want to keep .dtx files elsewhere.

\section*{E Acquiring the Software}

\subsection*{E.1 Repositories of \TeX material}

To aid the archiving and retrieval of \TeX-related files, a TUG working group developed the Comprehensive \TeX Archive Network (CTAN). Each CTAN site has identical material, and maintains authoritative versions of its material. These collections are extensive; in particular, almost everything mentioned in this article is archived at the CTAN sites, even if its location isn't explicitly stated.

The CTAN sites are currently \texttt{ftp.dante.de} (129.206.100.192), \texttt{ftp.tex.ac.uk} (128.232.1.87), and \texttt{ftp.shsu.edu} (192.92.115.10). The organisation of \TeX files on all these sites is identical and starts at \texttt{tex-archive/}. To reduce network load, please use the CTAN site or mirror closest to you. A complete and current list of CTAN sites and mirrors can be obtained by using the \texttt{finger} utility on `user' \texttt{ctan_us@ftp.SHSU.edu}; it is also available as file \texttt{CTAN.sites}.

To find software at a CTAN site, use anonymous \texttt{ftp} to the host, and then execute the command `quote site index <term>' (see question \ref{sec:findfonts} for details).

Email servers \texttt{ftpmail@ftp.shsu.edu} and \texttt{ftpmail@ftp.dante.de} provide an \texttt{ftp}-like interface through mail. Send a message containing just the line `help' to your nearest server for details of use.

Users on BITNET may access anonymous \texttt{ftp} for some files can be had indirectly by sending mail to \texttt{BITFTP@PUCC.BITNET}. Send a message containing the line `help' to this address for more information.

There is also the DECUS \TeX collection of material for VMS, Unix, MS-DOS, and the Macintosh. The material for VMS has not been kept up-to-date, but continues to run on OpenVMS on the VAX. It is available via anonymous \texttt{ftp} from \texttt{wuarchive.wustl.edu} (128.252.135.4) in \texttt{decus/tex/}. It can also be obtained from the DECUS Library (reference number VS0058) in the US, or through your DECUS office outside of the US. To contact the DECUS Library, send mail or telephone:

\begin{verbatim}
DECUS
LIBRARY ORDER PROCESSING
334 South Street, SHR3-1/T25
Shrewsbury, MA 01545-4195
USA
Tel: 800-DECUS55 (within the USA, for information)
Fax: (+1) 508-841-3373 (for inquiries)
\end{verbatim}

or send electronic mail for information to the DECUS \TeX Collection Editor, Ted Nieland (nieland@ted.hcst.com).

Finally, of course, the \TeX user who has no access to any sort of network may buy a copy of the archive on CD-ROM (see question \ref{sec:CD-ROM}).

\subsection*{E.2 Contributing a file to the archives}

Use anonymous \texttt{ftp} to any CTAN archive (see question \ref{sec:findfonts}) and retrieve the file \texttt{README.uploads} in the root directory. It contains instructions for uploading files and notifying the appropriate people for that site.

If you cannot use \texttt{ftp}, mail your contribution to \texttt{sty-mgr@shsu.edu} and it will be passed along. You will make everyone's life easier if you choose a descriptive and unique name for your submission, so it's probably a good idea to check that your style file's name is not already in use by means of the 'site index' command (see question \ref{sec:findfonts}).

\subsection*{E.3 Finding new fonts}

A comprehensive list of \MetaFont fonts is posted to \texttt{comp.fonts} and to \texttt{comp.text.tex}, roughly every six weeks, by Lee Quin (lee@sq.sq.com); it is available as \texttt{info/metafont-list}.

The list contains details both of commercial fonts and of fonts available via anonymous \texttt{ftp}. Most of the fonts are available via anonymous \texttt{ftp} from the CTAN archives (see question \ref{sec:findfonts}).
If you don’t have access to the Internet, you can get the CTAN collections on a CD-ROM. Even those who do will find it very convenient to have 600Mb of \TeX\-related files to hand. Prime Time Freeware produced \TeXcetera 1.1 in July 1994, which is a snapshot of CTAN taken in June 1994. Regular updates are planned. The material is all compressed in ZIP format to fit it all on one CD, and to avoid the limitations of the ISO 9660 file system directory. You can buy the CD from:

Prime Time Freeware
370 Altair Way, Suite 150
Sunnyvale CA 94086
USA
Tel: (+1) 408 433 9662
Fax: (+1) 408 433 0727
Email: ptf@cfcl.com

or from many CD-ROM resellers, or the TUG office (see question \textsuperscript{14}). Price will be around $60. Please note that PTF is not a big commercial firm, and is a good friend of the \TeX\ community.

If you want a ready-to-run \TeX\ system on CD-ROM, one is available for DOS only (so far). The Dutch \TeX\ Users Group (NTG) publish the whole 4All\TeX\ workbench on a CD-ROM packed with all the MS-DOS \TeX\ software, macros and fonts you can want. It is available from NTG direct (see question \textsuperscript{15}), from TUG for $40 and from UKTUG for £30 (a manual is included). It is a useful resource for anyone to browse, not just for DOS users.

\section{\TeX\ Systems}

\subsection{\TeX/\LaTeX\ for different machines}

\textbf{Unix} Instructions for retrieving the Unix \TeX\ distribution via anonymous ftp are available in the document systems/unix/unixtex.ftp

\textbf{AIX} \TeX\ for the IBM RS6000 running AIX is available in systems/unix/aix3.2

\textbf{386/ix} Executables for 386/ix are available in systems/unix/386ix

\textbf{Linux} There are at least two fairly complete implementations of \TeX\ to run on Linux. The Slackware distribution includes \NTeX\ (available as systems/unix/linux/ntex), which probably contains more \TeX\-related material than you would ever want. The more recent \TeTeX\ (available as systems/unix/linux/teTeX) is based on Karl Berry’s path-searching mechanisms, and is more compact than \NTeX\ while still being pretty comprehensive.

\textbf{PC} The \emTeX\ package for PCs running MS-DOS or OS/2 includes \LATEX, \BIBTEX, previewers, and drivers, and is available in systems/msdos/emtex as a series of disk images. The package was written by Eberhard Mattes, and documentation is available in both German and English. Windows users, or those wishing to run current \LaTeX\ on machines with a 386 or better processor, should get the beta-test release of \emTeX\ (systems/msdos/emtex/betatest), and the RSX memory extender (systems/msdos/dpmigcc).

A second package, \gTeX, runs under MS-DOS or Windows (and its users speak well of it). It is available from systems/msdos/gtex

TUG (and some of the other user groups) offer all freely-available \TeX\ software for the PC. A catalogue is available free from TUG (see question \textsuperscript{14}).

\textbf{Mac} Oz\TeX\ is a shareware version of \TeX\ for the Macintosh. A dvi previewer and PostScript driver are also included. It should run on any Macintosh Plus, SE, II, or newer model, but will not work on a 128K or 512K Mac. It was written by Andrew Trevorrow, and is available in systems/mac/oztex, or on floppy disks from TUG (see question \textsuperscript{14}). UK TUG prepays the shareware fee, so that members of UK TUG may acquire the software without further payment. Questions about Oz\TeX\ may be directed to oztex@midway.uchicago.edu

Another partly shareware program is CMac\TeX\ (available as systems/mac/cmactex), put together by Tom Kiffe. This is much closer to the Unix \TeX\ setup (it uses dvips, for instance).

\textbf{VMS} \TeX\ for VMS is available as systems/vms/Alpha/tex_axp_exe.zip (for Alpha-based machines) or systems/vms/VAX/tex_vax_exe.zip (for VAX machines). Standard tape distribution is through DECUS (see question \textsuperscript{27}).

\textbf{Atari} \TeX\ is available for the Atari ST in systems/atari

If anonymous ftp is not available to you, send a message containing the line “help” to atari@atari. archive.umich.edu
32 \TeX{-friendly editors and shells}

There are good \TeX{-writing environments and editors for most operating systems; some are described below, but this is only a personal selection:

**Unix** Try GNU *emacs*, and the AUCTeX mode (support/auctex). This provides menu items and control sequences for common constructs, checks syntax, lays out markup nicely, lets you call \TeX{} and drivers from within the editor, and everything else like this that you can think of. Complex, but very powerful.

**VMS** An *Isedit* mode for editing \TeX{} source is available from TUG (see question[14]) as \TeX{}niques 1, VAX Language-Sensitive Editor, by Kent MacPherson (1985).

**MS-DOS**

- The 4All\TeX{} workbench (systems/msdos/4alltex) provides a very comprehensive environment written in 4DOS (you need to pay a shareware fee for this) which lets you access most \TeX{}-related software in a friendly way. You can choose your own editor; something such as QEdit or Brief is suitable. This whole package is available in easy-to-use form on CD-ROM from \TeX{} user groups.
- \TeX{}Xshell (systems/msdos/texshell) is a simpler, easily-customisable environment, which can be used with the editor of your choice.
- Eddi4\TeX{} (systems/msdos/e4t) is a specially-written \TeX{} editor which features intelligent colouring, bracket matching, syntax checking, online help and the ability to call \TeX{} programs from within the editor. It is highly customisable, and features a powerful macro language.

You can also use GNU *emacs* and AUCTeX under DOS.

**Windows** Your best public domain bet is probably to use MicroEmacs as an editor and control centre for \TeX{} programs. The \TeX{} package (systems/msdos/gtex) comes with MicroEmacs ready to go, integrated with \TeX{}, previewer, dvips and GhostScript.

On a PC with large enough memory, a version of GNU *emacs*, that will run under Windows, is available; thus you can also use AUCTeX under Windows.

Y\&Y’s commercial (and high-quality) Windows previewer, *dviwindo*, can be used as a good \TeX{} shell, calling programs such as \TeX{}, drivers, and editors (Y\&Y supply the public domain PE, and recommend the commercial Epsilon) from customisable menus (see question[33] for details of Y\&Y).

Scientific Word is a WYSIWYG editing program, strong on maths, which uses \LaTeX{} for output (see question[33] for contact address).

**OS/2** Eddi4\TeX{} works under OS/2; look also at systems/os2/epmtex for a specific OS/2 shell.

**Macintosh** The commercial Textures provides an excellent integrated Macintosh environment with its own editor. More powerful still (as an editor) is the shareware Alpha (systems/mac/support/alpha) which is extensible enough to let you perform almost any \TeX{}-related job. It works well with Oz\TeX{}.

Atari, Amiga and NeXT users also have nice environments. \LaTeX{} users who like ‘make’ can find help in, *e.g.*, support/latexmk

There is another set of shell programs to help you manipulate \BibTeX{} databases.

33 Commercial \TeX{} implementations

There are many commercial implementations of \TeX{}. The first appeared not long after \TeX{} itself appeared. Of the vendors, ArborText (formerly Textset) and Personal \TeX{} are those who have survived longest (since the mid or early 80s).

What follows is probably an incomplete list. Naturally, no warranty or fitness for purpose is implied by the inclusion of any vendor in this list. The source of the information is given to provide some clues to its currency.

In general, a commercial implementation will come ‘complete’, that is, with suitable previewers and printer drivers. They normally also have extensive documentation (*i.e.*, not just the \TeX{}book!) and some sort of support service. In some cases this is a toll free number (probably applicable only within the USA and or Canada), but others also have email, and normal telephone and fax support.
Unix; \TeX\ Silicon Graphics Iris/Indigo, Solaris 2.1, IBM RS/6000, DEC/RISC-Ultrix, HP 9000. “Complete \TeX\ packages. Ready to use, fully documented and supported.”

ArborText Inc
1000 Victors Way
Suite 400
Ann Arbor MI 48108
USA
Tel: (+1) 313-996-3566
Fax: (+1) 313-996-3573

Source: \textit{TUG}boat, 1994, 15(1)

VAX/VMS; \textbf{Convergent \TeX}\ Complete system for VAX/VMS machines (a version for Alphas is in preparation); includes \LaTeX, multinational typesetting support, \textsc{Metafont} and Web.

Northlake Software, Inc.
812 SW Washington, Ste 1100
Portland, OR 97201
USA
Tel: (+1) 503-228-3383
Fax: (+1) 503-228-5662
Email: rau@nls.com

Source: Email from Pat Rau, November 1994

PC; \textbf{Turbo\TeX}\ Runs on Windows, MS-DOS, OS/2, Unix and VAX/VMS; also True\TeX\ for Windows 3.1, Window NT and Windows 95.

The Kinch Computer Co.
611 Mitchell Street
Ithaca, New York 14850
USA
Tel: (+1) 607-273-0222
Fax: (+1) 607-273-0484
Email: kinch@netcom.com

Source: AMS listing, November 1994

PC; \textbf{\TeX}\ “Bitmap free \TeX\ for Windows.”

Y&Y, Inc.
45 Walden Street
Concord MA 01742
USA
Tel: 800-742-4059 (within the USA)
Tel: (+1) 508-371-3286
Fax: (+1) 508-371-2004

Source: AMS listing, November 1994

pc\TeX\ Long-established: now has a Windows implementation.

Personal \TeX\ Inc
12 Madrona Street
Mill Valley, CA 94941
USA
Fax: (+1) 415-388-8865
Email: pti@crl.com

Source: \textit{TUG}boat, 1992, 13(2)

PC; \textbf{VT\TeX}\ Also “Bitmap-free”.

MicroPress Inc
68-30 Harrow Street
Forest Hills, NY 11375
The New \TeX\ FAQ

USA
Tel: (+1) 718-575-1816
Fax: (+1) 718-575-8038

Source: Resource Directory supplement to TUGboat 1991, 12(12)

PC; micro\TeX\ Micro\TeX\ and \TeX\ tools.
Micro Programs, Inc.
251 Jackson Ave.
Syosset, NY 11791
USA
Tel: (+1) 516-921-1351
Email: sales@microprograms.com

Source: AMS listing, November 1994

PC; Scientific Word Scientific Word and Scientific Workplace offer a mechanism for near-WYSIWYG input of \LaTeX\ documents; they ship with True\TeX\ from Kinch (see above). Queries within the UK should be addressed to Scientific Word Ltd., others should be addressed directly to the publisher, TCI.

Dr Christopher Mabb
Scientific Word Ltd.
98 Pont Adam
Ruabon
Wrexham
Clwyd, LL14 6EF
UK
Tel: 01978 823088 (within the UK)
Fax: 01978 823066 (within the UK)
Email: christopher@sciword.demon.co.uk

TCI Software Research Inc.
1190 Foster Road
Las Cruces
New Mexico, 88001
USA
Tel: (+1) 505-522-4600
Fax: (+1) 505-522-0116
Email: info@tcisoft.com

Source: Mail from Christopher Mabb, November 1994

Macintosh; Textures “A \TeX\ system ‘for the rest of us’”; also gives away a METAFONT implementation and some font manipulation tools.
Blue Sky Research
534 SW Third Avenue
Portland, Oregon 97204
USA
Tel: 800-622-8398 (within the USA)
Tel: (+1) 503-222-9571
Fax: (+1) 503-222-1643
Email: sales@bluesky.com

Source: TUGboat, 1994, 15(1)

Amiga\TeX\ A full implementation for the Commodore Amiga, including full, on-screen and printing support for all PostScript graphics and fonts, IFF raster graphics, automatic font generation, and all of the standard macros and utilities.
Radical Eye Software
PO Box 2081
Stanford, CA 94309
USA
G  DVI Drivers and Previewers

34  DVI to PostScript conversion programs

The best public domain dvi to PostScript conversion program which runs under many operating systems is Tom Rokicki’s dvips. dvips is written in C and ports easily to other operating systems; it is available as dviware/dvips

VMS versions are available through the DECUS library (see question 27), and also from CTAN: systems/vms/Alpha/dvips555_arp_exe.zip (for Alpha-based machines), systems/vms/VAX/dvips555_vax_exe.zip (for VAX machines); support files are available in systems/vms/dvips555_support.zip, and a set of fonts for use with \LaTeX2e are available in systems/vms/dvips_fonts_support.zip

A precompiled version for MS-DOS is available from systems/msdos/dviware/dvips

Karl Berry’s version of dvips (called dvipsk) has a configure script and path searching code similar to that in his other programs (e.g., web2c); it is available from dviware/dvipsk

Another good portable program is dvitops by James Clark, which is also written in C and will compile under Unix, MS-DOS, VMS, and Primos; however, it does not support virtual fonts. It is available from dviware/dvitops

Macintosh users can use either the excellent drivers built into Oz\TeX or Textures, or a port of dvips in the CMac\TeX package.

35  DVI drivers for HP LaserJet

The em\TeX package (see question 31) contains a driver for the LaserJet, dvihplj.

Version 2.10 of the Beebe drivers supports the LaserJet. These drivers will compile under Unix, VMS, and on the Atari ST and DEC-20’s, and are available from dviware/beebe

Karl Berry’s dviljk, which has the same path-searching code as his dvipsk (see question 34), is available in dviware/dviljk

36  DVI previewers

Em\TeX and g\TeX for the PC, and Oz\TeX for the Macintosh, all come with previewers that can be used on those platforms. There is a good OS/2 Presentation Manager previewer in em\TeX, and a public domain Windows previewer (dviware/dviwin). Commercial PC \TeX packages (see question 33) have good DOS and Windows previewers.

Previewers available for other operating systems include:

xdvi  The most widely used previewer for the X Window System (and hence almost any Unix or modern VMS workstation); available in dviware/xdvi

Karl Berry’s version of xdi, called xdvik, has features analogous to his dvipsk (see question 34); it is available in dviware/dxvik

dvipage  For SunView on (old enough) Sun workstations. This was published in volume 15 of comp.sources.unix and is archived in dviware/dvipage

xtex  An older previewer for the X Window System; available in dviware/seetex

dviapollo  For Apollo Domain workstations; available in dviware/dviapollo

dvidis  For (old enough, VMS) VAXstations running VWS; available in dviware/dvidis

dvitovdu  for Tektronix 4010-compatible and other terminals under Unix and VMS; available as dviware/dvitovdu

dvi2tty  A dvi to ASCII conversion program, for normal terminals; available as dviware/dvi2tty

texsgi  For SGI under Irix; both a binary and source are available, but be sure to get the fonts as well. Available as dviware/texsgi

H  Support Packages for \TeX

37  Fig, a \TeX-friendly drawing package

(X)Fig is a menu driven tool that allows you to draw objects on the screen of an X workstation. transfig is a set of tools which translate the code fig produces to other graphics languages including PostScript and the \LaTeX picture environment. They are available in graphics/xfi and graphics/transfig

Fig is supported by Micah Beck (beck@c.s.cornell.edu) and transfig is maintained by Brian Smith (bvsmith@lbl.gov). Another tool for fig conversion is fig2mf which generates METAFONT code from fig input. It is available in graphics/fig2mf
38 \TeXCAD, a drawing package for \TeX

\TeXCAD is a program for the PC which enables the user to draw diagrams on screen using a mouse or arrow keys, with an on-screen menu of available picture-elements. Its output is code for the \LaTeX picture environment. Optionally, it can be set to include lines at all angles using the \em \TeX driver-family \specials (see question 25). \TeXCAD is part of the \em \TeX distribution.

A Linux port of the program, systems/unix/linux/xtexcad-2.1.tar.z, is reported also to run on other Unix operating systems.

39 Spelling checkers for work with \TeX

For Unix, \texttt{ispell} is probably the program of choice. It is available in support/ispell; beware of any version with a number 4.x — such versions represent a divergent version of the source which lacks many useful facilities of the 3.x series.

For MS-DOS, there are several programs. \texttt{amspell} can be called from within an editor (available as support/amspell). \texttt{jspell} is an extended version of \texttt{ispell} (available as support/jspell).

For the Macintosh, \texttt{Excalibur} is the program of choice. It will run in native mode on both sorts of Macintosh, and is available as systems/mac/support/excalibur/Excalibur-20-sea.hqx (there are other dictionaries in the same directory).

For VMS, a spell checker can be found in support/vmspell.

40 The \Vortex\ package

\Vortex (available in support/vortex) is a package of programs written at the University of California at Berkeley, and was described by Michael A. Harrison in “News from the \Vortex project” in TUGboat 10(1), pp. 11–14, 1989. It includes several nice previewers and some \texttt{emacs} modes for \TeX and \BIB\TeX. The \Vortex distribution is not maintained, and now looks distinctly long in the tooth (it was never upgraded to \TeX version 3).

I Literate programming

41 What is Literate Programming?

Literate programming is the combination of documentation and source together in a fashion suited for reading by human beings. In general, literate programs combine source and documentation in a single file. Literate programming tools then parse the file to produce either readable documentation or compilable source. The WEB style of literate programming was created by D. E. Knuth during the development of his \TeX typesetting software.

Discussion of literate programming is conducted in the newsgroup \texttt{comp.programming.literate}, which is gatewayed to the mailing list \texttt{litprog@shsu.edu} (see question 19 for details). The literate programming FAQ is stored as \texttt{help/LitProg-FAQ}.

42 WEB for C, FORTRAN, and other languages

\TeX is written in the programming language WEB; WEB is a tool to implement the concept of “literate programming”.

\texttt{CWEB}, a WEB for C programs, written by Silvio Levy, is available as web/c_cpp/cweb.

Spidery WEB supports many languages including Ada, awk, and C. It was written by Norman Ramsey and, while not in the public domain, is usable free. It is available in web/spiderweb.

\texttt{FWEB} is a version for Fortran, Ratfor, and C written by John Krommes. It is available in web/fweb.

\texttt{SchemeWEB} is a Unix filter that translates SchemeWEB into \LaTeX source or Scheme source. It was written by John Ramsdell and is available in web/schemeweb.

\texttt{APLWEB} is a version of WEB for APL and is available in web/apl/aplweb.

\texttt{FunnelWEB} is a version of WEB that is language independent. It is available in web/funnelweb.

Other language independent versions of WEB are \texttt{nuweb} (which is written in ANSI C), available in web/nuweb, and \texttt{noweb}, available in web/noweb.

J Format conversions

43 Conversion between \TeX/\LaTeX and others

\texttt{troff/troff-to-latex}, written by Kamal Al-Yahya at Stanford University (California, USA), assists in the translation of a \texttt{troff} document into \LaTeX format. It recognises most –\texttt{ms} and –\texttt{man} macros, plus most \texttt{eqn} and some \texttt{tbl} preprocessor commands. Anything fancier needs to be done by hand. Two style files are provided. There is also a man page (which converts very well to \LaTeX…). The program is copyrighted but free. An enhanced version of this program, \texttt{tr2latex}, is available in support/tr2latex.
The DECUS \TeX{} distribution (see question 27) also contains a program which converts troff to \TeX{}.

**Scribe** Mark James (jamesm@dialogic.com) has a copy of scribe2latex he has been unable to test but which he will let anyone interested have. The program was written by Van Jacobson of Lawrence Berkeley Laboratory.

**WordPerfect** wp2latex (available as support/wp2latex) is a PC program written in Turbo Pascal by R. C. Houtepen at the Eindhoven University in the Netherlands. It converts WordPerfect 5.0 documents to \LaTeX{}. Pascal source is included. Users find it “helpful” and “decent” in spite of some limitations. It gets high marks for handling font changes, but cannot make indices, tables of contents, margins or graphics, and can’t handle features new in WordPerfect version 5.1, in particular the equation formatter. The program is copyrighted but free.

Glenn Geers of the University of Sydney, Australia (glenn@qed.physics.su.oz.au) is translating wp2latex into C and adding some WordPerfect 5.1 features, in particular its equation handling. His work is in the glenn subdirectory of support/wp2latex.

**PC-Write** pcwritex.arc is available in support/pcwritex. This is a print driver for PC-Write that “prints” a PC-Write V2.71 document to a \TeX{}-compatible disk file. It was written by Peter Flynn at University College, Cork, Republic of Ireland.

**runoff** Peter Vanroose’s (vanroose@esat.kuleuven.ac.be) conversion program is written in VMS Pascal. The sources and a VAX executable are available from support/rnototex.

**refer/tib** There are a few programs for converting bibliographic data between BIB\TeX{} and refer/tib formats. They are in biblio/bibtex/utils/refer-tools. In spite of the directory name, it also contains a shell script to convert BIB\TeX{} to refer as well. The collection is not maintained.

**RTF** A program for converting Microsoft’s Rich Text Format to \TeX{} is available in support/rtf2tex, which was written and is maintained by Robert Lupton (rh1@astro.princeton.edu). There is also a convertor to \LaTeX{} by Erwin Wecht, in support/rtf2latex.

**Microsoft Word** A rudimentary program for converting MS-Word to \LaTeX{} is wd2latex, for MS-DOS (dviware/wd2latex); a better idea, however, is to convert the document to RTF format and use one of the RTF converters mentioned above.

A group at Ohio State University (USA) is working on a common document format based on SGML, with the ambition that any format could be translated to or from this one. Framemaker provides “import filters” to aid translation from alien formats (presumably including \TeX{}), to Framemaker’s own.

### 44 Conversion from \TeX{}/\LaTeX{} to plain ASCII

The aim here is to emulate the Unix nroff, which formats text as best it can for the screen, from the same input as the Unix typesetting program troff.

Ralph Droms (droms@bucknell.edu) has a style file and a program that provide the \LaTeX{} equivalent of nroff, though it doesn’t do a good job with tables and mathematics. The software is available in support/txt; the original dvi2tty often does an acceptable job and is available in dviware/dvi2tty.

Another possibility is to use screen.sty (available as macros/latex209/contrib/misc/screen.sty). Use a dvi2tty program of some kind; you might try dviware/crudetype as well. Another possibility is to use the \LaTeX{}-to-ASCII conversion program, l2a (support/l2a), although this is really more of a de-\TeX{}ing program.

The canonical de-\TeX{}ing program is detex (support/detex), which removes all comments and control sequences from its input before writing it to its output. Its original purpose was to prepare input for a dumb spelling checker.

### 45 Conversion from HTML or SGML to \TeX{}

Nathan Torkington (Nathan.Torkington@vuw.ac.nz) developed html2latex from the HTML parser in NCSA’s Xmosaic package. The program takes an HTML file and generates a \LaTeX{} file from it. The conversion code is subject NCSA restrictions, but the whole source is available as support/html2latex.

Peter Flynn has developed a pcl program sgml2tex (for PCs). It is controlled by a configuration file, and an example configuration for translating HTML files is given. The program is available in support/sgml2tex.

### 46 Conversion to HTML

\LaTeX{}2HTML (support/latex2html) is a package (mostly of perl scripts) that breaks up a \LaTeX{} document into one or more components, and links them together so that they can be read over the World-Wide Web as an hypertext document.

\LaTeX{}2HTML defines a mapping between \LaTeX{} intra-document references and hyperlinks, and extends the \LaTeX{}
The New \TeX\ FAQ

referencing mechanisms to permit reference to other (possibly remote) documents and other internet resources. It translates \TeX\ accented and other characters (as best it can) to things that World-Wide Web browsers can display, and provides a mechanism whereby mathematics (and other things that browsers can’t deal with) may be translated to images that can be loaded in-line into the hypertext document.

K METAFONT

47 Getting METAFONT to do what you want

METAFONT allows you to create your own fonts, and most \TeX\ users will never need to use it. METAFONT, unlike \TeX, requires some customisation: each output device for which you will be generating fonts needs a mode associated with it. Modes are defined using the mode_def convention described on page 94 of The \TeX\book (see question 17). You will need a file, which conventionally called local.mf, containing all the mode_defs you will be using. If local.mf doesn’t already exist, Karl Berry’s collection of modes, available as fonts/modes/modes.mf, is a good starting point (it can be used as a ‘local.mf’ without modification in a ‘big enough’ implementation of METAFONT). Lists of settings for various output devices are also published periodically in TUGboat (see question 14). Now create a plain base file using \texttt{inimf.plain.mf} and \texttt{local.mf}:

```plaintext
% inimf
This is METAFONT...
**plain  you type ‘plain’
(output)
*input local  you type this
(output)
*dump  you type this
Beginning to dump on file plain...
(output)
```

This will create a base file named plain.base (or something similar; for example, it will be PLAIN.BAS on MS-DOS systems) which should be moved to the directory containing the base files on your system (note that some systems have two or more such directories, one for each ‘size’ of METAFONT used).

Now you need to make sure METAFONT loads this new base when it starts up. If METAFONT loads the plain base by default on your system, then you’re ready to go. Under Unix (using the default \texttt{web2c} distribution) this does indeed happen, but we could for instance define a command \texttt{mf} which executes \texttt{virmf &plain} loading the plain base file.

The usual way to create a font with plain METAFONT is to start it with the line

```
\mode=<mode name>; mag=<magnification>; input <font file name>
```

in response to the ‘***’ prompt or on the METAFONT command line. (If \texttt{<mode name>} is unknown or omitted, the mode defaults to ‘proof’ and METAFONT will produce an output file called \texttt{<font file name>.2602gf}) The \texttt{<magnification>} is a floating point number or ‘magstep’ (magsteps are defined in \TeX\book and \texttt{The \TeX\book}). If \texttt{mag=<magnification>} is omitted, then the default is 1 (magstep 0). For example, to generate cmr10 at 12pt for an epson printer you would type

```
mf \mode=epson; mag=magstep 1; input cmr10
```

Note that under Unix the \texttt{\} and ; characters must usually be quoted or escaped, so this would typically look something like

```
mf ‘\mode=epson; mag=magstep 1; input cmr10’
```

If you don’t have \texttt{inimf} or need a special mode that isn’t in the base, you can put its commands in a file (e.g., ln03.mf) and invoke it on the fly with the \texttt{\smode} command. For example, to create cmr10.300gf for an LN03 printer, using the file

```
% This is ln03.mf as of 2/27/90
% mode_def courtesy of John Sauter
```

\footnote{The \texttt{command_name} is symbolically linked to \texttt{virmf}, and \texttt{virmf} loads \texttt{command_name.base}}
proofing:=0;
fontmaking:=1;
tracingtitles:=0;
pixels_per_inch:=300;
blacker:=0.65;
fillin:=-0.1;
o_correction:=.5;

(note the absence of the mode_def and enddef commands), you would type

mf \smode="ln03"; input cmr10

This technique isn’t one you should regularly use, but it may prove useful if you acquire a new printer and want to experiment with parameters, or for some other reason are regularly editing the parameters you’re using. Once you’ve settled on an appropriate set of parameters, you should use them to rebuild the base file that you use.

A summary of the above written by Geoffrey Tobin, and tips about common pitfalls in using METAFONT, is available as documentation/metafont_for_beginners.tex

48 Which font files should be kept

METAFONT produces from its run three files, a metrics (tfm) file, a generic font (gf) file, and a log file; all of these files have the same base name as does the input (e.g., if the input file was cmr10.mf, the outputs will be cmr10.tfm, cmr10.nnngf, and cmr10.log).

To operate TeX using the font, you need a tfm file, so you need to keep that. However, you are likely to want generate the same font at more than one magnification, and each time you do so you’ll (incidentally) generate another tfm file; these files are all the same, and you only therefore need to retain one of them.

To produce preview or printed output, the dvii processor will need a font raster file; this is what the gf file provides. However, while there used (once upon a time) to be dvii processors that could use gf files, modern processors use packed raster (pk) files. Therefore, you need to generate a pk file from the gf file; the program gftopk does this for you, and once you’ve done that you may throw the gf file away.

The log file should never need to be used, unless there was some sort of problem in the METAFONT run, and need not be ordinarily kept.

49 Getting bitmaps from the archives

Most people these days start using TeX with a 300 dots-per-inch (dpi) laser printer, and Computer Modern bitmap fonts for this resolution are supplied with most TeX packages. There are also two such sets available on CTAN: fonts/cm/pk/pk300.zip (for write-black printer engines) and fonts/cm/pk/pk300w.zip (for write-white engines). However, some users want to send their work to high quality typesetting machines (typically with a resolution of 1270 dpi or greater); it is also becoming more common to use a 600 dpi laser printer. Why don’t the archives or suppliers provide bitmap fonts at these sizes? There are two reasons:

1. When a bitmap font is created with METAFONT, it needs to know the characteristics of the device; who knows what 600 or 1270 dpi device you have? (Of course, this objection applies equally well to 300 dpi printers.)
2. Bitmap fonts get big at high resolutions. Who knows what fonts at what sizes you need?

It would be possible to provide some set of 1270 dpi bitmap fonts in the archives, but it would take a lot of space, and might not be right for you.

So what to do? You can build the fonts you need yourself with METAFONT: this isn’t at all hard, and some drivers help you (dvips, and the emTeX drivers) construct the METAFONT commands. You might need to look at Karl Berry’s collection of METAFONT modes (fonts/modes/modes.mf). Alternatively, if it is a PostScript device you have, consider using the fonts in Type 1 font format. You can buy all the Computer Modern fonts in PostScript form from Blue Sky Research or Y&Y (see question 33 for addresses), or you can use Basil Malyshev’s public domain versions in fonts/cm/ps-type1 (the Paradissa collection is complete, but is gradually being replaced by the better BaKoMa collection).

L PostScript and TeX

\footnote{Note that the file name may be transmuted by such operating systems as MS-DOS, which don’t permit long file names}
50  Using PostScript fonts with \TeX

Before you start: in order to use PostScript fonts, \TeX needs metric (called tfm) files. Several sets of metrics are available from the archives; for mechanisms for generating new ones, see question 52. You also need the fonts themselves; PostScript printers come with a set of fonts built in, but to extend your repertoire you almost invariably need to buy from one of the many commercial font vendors (see, for example, question 53).

If you use \LaTeXe, the best way to get PostScript fonts into your document is to use the PSNFSS package maintained by Sebastian Rahtz and Alan Jeffrey (available in macros/latex/packages/psnfss); it’s supported by the \LaTeXe project team, so bug reports can and should be submitted. PSNFSS gives you a set of packages for changing the default roman, sans-serif and typewriter fonts; e.g., \texttt{times.sty} will set up Times Roman, Helvetica and Courier in place of Computer Modern, while \texttt{avant.sty} just changes the sans-serif family to AvantGarde. To go with these packages, you will need the font metric files (watch out for encoding problems! see question 52 and font description (.fd) files for each font family you want to use. These can be obtained from fonts/metrics, arranged by vendor (e.g., Adobe, Monotype, etc.). For convenience, metrics for the common ‘35’ PostScript fonts found in most printers are provided with PSNFSS, packaged as macros/latex/packages/psnfss/lw35nfss.zip.

For older versions of \LaTeX there are various schemes, of which the simplest to use is probably the PS\LaTeX macros distributed with dvips.

For \TeX, you load whatever fonts you like; if the encoding of the fonts is not the same as Computer Modern it will be up to you to redefine various macros and accents, or you can use the font re-encoding mechanisms available in many drivers and in \texttt{ps2pk} and \texttt{afm2tfm}.

Victor Eijkhout’s sophisticated Lollipop package (macros/lollipop) supports declaration of font families and styles in a similar way to \LaTeX’s NFSS, and so is easy to use with PostScript fonts.

Some common problems encountered are discussed elsewhere (see question 53).

51  Previewing files using PostScript fonts

Most \TeX previews only display bitmap \texttt{pk} fonts. If you want to preview documents using PostScript fonts, you have three choices:

1. Convert the .dvi file to PostScript and use a PostScript previewer. Some modern Unix X implementations have this built in (as does NeXT-step); (X11) Unix, Windows, OS/2, and MS-DOS users can use the free GhostScript \texttt{(support/ghostscript)}, a complete level 2 implementation.

2. Under Windows on a PC, or on a Macintosh, let Adobe Type Manager display the fonts. Textures (Macintosh) works like this, and under Windows you can use Y&Y’s \texttt{dviwindo} for bitmap-free previewing. (See question 53 for details of these suppliers.)

3. If you have the PostScript fonts in Type 1 format, use \texttt{ps2pk} (fonts/utilities/ps2pk) or \texttt{gsftopk} (designed for use with the \texttt{ghostscript} fonts; fonts/utilities/gsftopk) to make \texttt{pk} bitmap fonts which your previewer will understand. This can produce excellent results, also suitable for printing with non-PostScript devices. Check the legalities of this if you have purchased the fonts. The very commonest PostScript fonts such as Times and Courier come in Type 1 format on disk with Adobe Type Manager (often bundled with Windows, and part of OS/2).

52  \TeX font metric files for PostScript fonts

Font vendors such as Adobe supply metric files for each font, in \texttt{afm} (Adobe Font Metric) form; these can be converted to \texttt{tfm} (\TeX Font Metric) form. The CTAN archives have prebuilt metrics which will be more than enough for many people (fonts/metrics; beware — this directory is at the root of a huge tree), but you may need to do the conversion yourself if you have special needs or acquire a new font. One important question is the encoding of (Latin character) fonts; while we all more or less agree about the position of about 96 characters in fonts (the basic ASCII set), the rest of the (typically) 256 vary. The most obvious problems are with floating accents and special characters such as the ‘pounds sterling’ sign. There are three ways of dealing with this: either you change the \TeX macros which reference the characters (not much fun, and error-prone); or you change the encoding of the font (easier than you might think); or you use virtual fonts (see question 24) to pretend to \TeX that the encoding is the same as it is used to. If you use \LaTeXe, it allows for changing the encoding in \TeX; read the \LaTeX Companion (see question 17) for more details. In practice, if you do much non-English (but Latin script) typesetting, you are strongly recommended to use the \texttt{fontenc} package with option ‘T1’ to select T1 (also known as ‘Cork’ or ‘DC’) encoding. The PSNFSS package (see question 50) provides support for PostScript (using virtual fonts) in this encoding.

Alan Jeffrey’s \texttt{fontinst} package (fonts/utilities/fontinst) is an \texttt{afm} to \texttt{tfm} converter written in \TeX; it is used to generate the files used by \LaTeXe’s PSNFSS package to support use of PostScript fonts. It is a sophisticated
package, not for the faint-hearted, but is powerful enough to cope with most needs. Much of its power relies on the
device driver that you are using PostScript fonts; in the case of dvips, this means adding lines to the
psfonts.map file. Otherwise, dvips will try to find pk files. If the font isn’t built into the printer, you have to acquire
it (in many cases this means buying it from a commercial supplier!). You then have to instruct the driver to download
it with each job (the mechanism depends on your driver). So it’s no good just installing the metrics for Optima and
expecting it to work. You have to pay hard cash for the font itself, which will come (for Unix and MS-DOS users) in
pfb (Printer Font Binary) form.

Second, you cannot expect your previewer to suddenly start displaying PostScript fonts; most of them only know
about PK bitmap fonts such as Computer Modern. You can create these from PostScript fonts if you have bought the
pfb file by using ps2pk (fonts/utilities/ps2pk); this would also let you use the fonts with non-PostScript de-
vice drivers such as the emTeX ones. It is your responsibility to make sure you are not breaking the licence restrictions
on your font if you bought it.

Third, the stretch and shrink between words is a function of the font metric; it is not specified in afm files, so
different converters choose different values. The PostScript metrics that come with PSNFSS do quite tight setting,
which may produce a lot of hyphenation or overfull boxes if you are not careful. If you are having trouble, Alan
Jeffrey recommends the following settings in a package file:

```latex
\tolerance800
\emergencystretch2em
\doublehyphenpenalty5000
\hbadness0pt
\leftskip0pt minus 1pt
\rightskip0pt minus 1pt
```

A future release of these fonts will implement looser typesetting by default, so these settings should become unnec-

54 Choice of scalable outline fonts

If you are interested in text alone, you can use any of over 20,000 fonts(!) in Adobe Type 1 format (called ‘PostScript
types’ in the TeX world and ‘ATM fonts’ in the DTP world), or any of several hundred fonts in TrueType format. That
is, provided of course, that your previewer and printer driver support scalable outline fonts.

TeX itself only cares about metrics, not the actual character programs. You just need to create a TeX metric file
.tfm using some tool such as afm2tfm, afmtotfm (from Y&Y, see question 33 or fontinst). For the previewer or printer
driver you need the actual outline font files themselves (pfa for Display PostScript, pfb for ATM on IBM PC, Mac
outline font files on Macintosh).

If you also need mathematics, then you are severely limited by the demands that TeX makes of maths fonts (for
details, see the paper by B.K.P. Horn in the Proceedings of the 1993 TUG conference at Aston University, published
in TUGboat, 1994, 14(3)). For maths, then, there are relatively few choices:

**Computer Modern** (75 fonts — optical scaling) Donald E. Knuth

Note that CM is available in scalable outline form. There are commercial as well as public domain versions, and
there are both Adobe Type 1 and TrueType versions. Some of these are ‘commercial grade,’ with full hand-tuned
hinting, some render very poorly, while others are merely incompatible with Adobe Type Manager (ATM).

**Lucida Bright with Lucida New Math** (25 fonts) Chuck Bigelow and Kris Holmes

Lucida is a family of related fonts including serifed, sans serif, sans serif fixed width, calligraphic, blackletter, fax,
Kris Holmes’ connected handwriting font, etc; they’re not as ‘spindly’ as Computer Modern, with a large x-height,
and include a larger set of maths symbols, operators, relations and delimiters than CM (over 800 instead of 384:
among others, it also includes the AMS \texttt{msam} and \texttt{msbm} symbol sets). The planned ‘Lucida Bright Expert’ (14 fonts) adds serifed fixed width, another handwriting font, smallcaps, bold maths, upright ‘maths italic’, \textit{etc.}, to the set. The distribution includes support for use with \texttt{plain \LaTeX} and \texttt{LaTeX\ 2.09}. Support under \texttt{LaTeX\ 2e} is provided in PSNFSS (see question\[50\] thanks to Sebastian Rahtz.

\textbf{MathTime 1.1 (3 fonts) \TeXplorators (Michael Spivak)}

The set contains maths italic, symbol, and extension fonts, designed to work well with Times-Roman. These are typically used with Times, Helvetica and Courier (which are resident on many printers, and which are supplied with some PC versions). In addition you may want to complement this basic set with Adobe’s Times Smallcap, and perhaps the set of Adobe ‘Math Pi’ fonts, which include blackboard bold, blackletter, and script faces. The distribution includes support for use with \texttt{plain \LaTeX} and \texttt{LaTeX\ 2.09} (including code to link in Adobe Math Pi 2 and Math Pi 6). Support under \texttt{LaTeX\ 2e} is provided in PSNFSS (see question\[50\] thanks to Sebastian Rahtz.

\textit{Adobe Lucida, LucidaSans and LucidaMath} (12 fonts)

Lucida and LucidaMath are generally considered to be a bit heavy. The three maths fonts contain only the glyphs in the CM maths italic, symbol, and extension fonts. Support for using LucidaMath with \TeX{} is not very good; you will need to do some work reencoding fonts \textit{etc.} (In some sense this set is the ancestor of the LucidaBright plus LucidaNewMath font set.)

\textit{Concrete, the AMS maths fonts \textit{etc.}} Donald E. Knuth and the AMS.

These are sometimes mentioned as alternatives to CM, but they are really adjuncts, in that you need to use at least the basic CM maths fonts with them.

\textbf{Proprietary fonts} Various sources.

Since having a high quality font set in scalable outline form that works with \TeX{} can give a publisher a real competitive advantage, there are some publishers that have paid (a lot) to have such font sets made for them. Unfortunately, these sets are not available on the open market, despite the likelihood that they’re more complete than those that are.

\textit{Mathptm} (4 fonts) Alan Jeffrey.

This set contains maths italic, symbol, extension, and roman virtual fonts, built from Adobe Times, Symbol, Zapf Chancery, and the Computer Modern fonts. The Mathptm fonts are free, and the resulting PostScript files can be freely exchanged. Contains most of the CM math symbols. Support under \texttt{LaTeX\ 2e} in PSNFSS (see question\[50\]) thanks to Alan Jeffrey and Sebastian Rahtz.

All of the first three font sets are available in formats suitable for IBM PC/Windows, Macintosh and Unix/NeXT from Y\&Y and from Blue Sky Research (see question\[33\] for details). The MathTime fonts are also available from:

\begin{center}
\TeXplorators
1572 West Gray #377
Houston TX 77019
USA
\end{center}

The very limited selection of maths font sets is a direct result of the fact that a maths font has to be explicitly designed for use with \TeX{} and as a result it is likely to lose some of its appeal in other markets. Furthermore, the \TeX{} market for commercial fonts is minute (in comparison, for example, to Microsoft TrueType font pack #1, which sold something like 10 million copies in a few weeks after release of Windows 3.1!).

Text fonts in Type 1 format are available from many vendors including Adobe, Monotype, Bitstream. Avoid cheap rip-offs: not only are you rewarding unethical behaviour, destroying the cottage industry of innovative type design, but you are \textit{also} very likely to get junk. The fonts may not render well (or at all under ATM), may not have the ‘standard’ complement of 228 glyphs, or may not include metric files (needed to make TFM files). Also, avoid TrueType fonts from all but the major vendors. TrueType fonts are an order of magnitude harder to ‘hint’ properly than Type 1 fonts and hence TrueType fonts from places other than Microsoft and Apple may be suspect. In any case you may find other problems with TrueType fonts such as service bureaux not accepting jobs calling for them.

55 \textit{Including a PostScript figure in \LaTeX{} 2e}\[25\]

\LaTeX\ 2e (see question\[55\]) has a standard package for graphics inclusion, rotation, colour, and other driver-related features. The package is documented in the second edition of the Lamport’s \LaTeX\ book (see question\[17\]), and is available in \texttt{macros/latex/packages/graphics}\[\texttt{psfig}\]

If you don’t use \LaTeX\ 2e, perhaps the best method is to use the \texttt{psfig} macros written by Trevor Darrell, available in \texttt{graphics/psfig}.

You will also need a \texttt{dvi} to PostScript conversion program that supports the \texttt{\special{s}}. The drivers mentioned

–25–
in question 34 do, and come with a version of \texttt{psfig} ready to use with them. The \texttt{psfig} macros work best with Encapsulated PostScript Files (EPS). In particular, \texttt{psfig} will need the file to have a BoundingBox (see Appendix H of the \textit{PostScript Language Reference Manual}). If you don’t have an EPS file, life can be difficult.

One point to note about including PostScript figures is that they are not part of the \texttt{dvi} file, but are only included when you use a \texttt{dvi} to PostScript conversion program. As a result, most \texttt{dvi} previewers will simply show the blank space \TeX has reserved for your figure, not the figure itself.

Anil K. Goel has written a long document describing in detail how to include figures, pictures, and images in \LaTeX documents, which is available as \texttt{info/figsinltx.ps}

\section{Special sorts of typesetting}

\subsection{Double-spaced documents in \LaTeX}

Are you producing a thesis, and trying to obey regulations that were drafted in the typewriter era? Or are you producing copy for a journal that insists on double spacing for the submitted articles?

\LaTeX is a typesetting system, so the appropriate design conventions are for “real books”. If your requirement is from thesis regulations, find whoever is responsible for the regulations, and try to get the wording changed to cater for typeset theses \textit{(e.g., to say “if using a typesetting system, aim to make your thesis look like a well-designed book”).} (If your requirement is from a journal, you’re probably even less likely to be able to get the rules changed, of course.)

If you fail to convince your officials, or want some inter-line space for copy-editing:

\begin{itemize}
  \item Try changing \begin{verbatim}
  \renewcommand{\baselinestretch}{1.2}
\end{verbatim}
  may be enough to give officials the impression you’ve kept to their regulations. Don’t try changing \begin{verbatim}
  \baselineskip
\end{verbatim} its value is reset at any size-changing command.
  \item Alternatively, use a line-spacing package. Options available are:
    \begin{itemize}
      \item for simple double spacing, \texttt{macros/latex209/contrib/misc/doublespace.sty},
      \item for greater flexibility, \texttt{macros/latex/contrib/other/misc/setspace.sty}, which has been upgraded for \LaTeX\texttt{2\epsilon}.
    \end{itemize}
\end{itemize}

\subsection{Formatting a thesis in \LaTeX}

Thesis styles are usually very specific to your University, so it’s usually not profitable to ask around for a package outside your own University. Since many Universities (in their eccentric way) still require double-spacing, you may care to refer to question 56. If you want to write your own, a good place to start is the University of California style (available as \texttt{macros/latex209/contrib/ucthesis}), but it’s not worth going to a lot of trouble. (If officials won’t allow standard typographic conventions, you won’t be able to produce an aesthetically pleasing document anyway!)

\subsection{Including a file in verbatim in \LaTeX}

A good way is to use Rainer Schöpf’s \texttt{verbatim.sty}, which provides the command \begin{verbatim}
  \verbatiminput{a file name}
\end{verbatim} that takes a file name as argument. This package is available as part of \texttt{macros/latex/packages/tools/}

Another way is to use the \texttt{alltt} environment, which requires the package available in \texttt{macros/latex209/contrib/misc/alltt.sty}

\subsection{Generating an index in \TeX/\LaTeX}

Making an index is not trivial; what to index, and how index it, is difficult to decide, and uniform implementation is difficult to achieve. You will need to mark all items to be indexed in your text (typically with \begin{verbatim}
  \index
\end{verbatim} commands).

It is not practical to sort a large index within \TeX, so a post-processing program is used to sort the output of one \TeX run, to be included into the document at the next run.

The following programs are available:

\begin{itemize}
  \item \texttt{makeindex} for \LaTeX under Unix (but runs under other OSs without changes). Available in \texttt{indexing/makeindex}; a version for the Macintosh is available as \texttt{systems/mac/makeindex.sit}, and ones for MS-DOS are part of the em\TeX and g\TeX distributions (the em\TeX version also runs under OS/2).
    The Makeindex documentation is a good source of information on how to create your own index. Makeindex can be used with some \TeX macro packages other than \LaTeX, such as Eplain.
  \item \texttt{idxtex} for \LaTeX under VMS. Available (together with a glossary-maker called \texttt{glotex}) in \texttt{indexing/glo+idxtex}
  \item \texttt{texindex} A witty little shell/\texttt{sed}-script-based utility for \LaTeX under Unix. Available from \texttt{support/texindex}
\end{itemize}

There are other programs called \texttt{texindex}, notably one that comes with the Texinfo distribution (see question 11).
60 Using $\LaTeX$ with plain $\TeX$

The file macros/eplain/btxmac.tex contains macros and documentation for using $\LaTeX$ with plain $\TeX$, either directly or with Eplain (see question [9]). See question [20] for more information about $\LaTeX$ itself.

61 Typesetting music in $\TeX$

A powerful package which allows the typesetting of orchestral and polyphonic music is Music$\TeX$, written by Daniel Taupin (taupin@rsovax.lps.u-psud.fr). It is available in macros/musictex

Digital music fans can typeset notation for their efforts by using midi2tex, which translates MIDI data files into Music$\TeX$ source code. It is available from support/midi2tex

A rather simpler notation than Music$\TeX$ is supported by abc2mtex; this is a package designed to notate tunes stored in an ASCII format (abc notation). It was designed primarily for folk and traditional tunes of Western European origin (such as Irish, English and Scottish) which can be written on one stave in standard classical notation. However, it should be extendable to many other types of music. It is available from support/abc2mtex

There is a mailing list for discussion of typesetting music in $\TeX$. To subscribe, send mail to mutex-request@stolaf.edu containing the word 'subscribe' in the body.

62 Drawing Feynman diagrams in $\LaTeX$

Michael Levine’s macro package for drawing Feynman diagrams in $\LaTeX$ is available in macros/latex209/contrib/feynman (see also macros/latex/contrib/supported/feynmf).

N How do I do $X$ in $\TeX$ or $\LaTeX$?

63 Proof environment

It is not possible to make a proof environment which automatically includes an ‘end-of-proof’ symbol. Some proofs end in displayed maths; others do not. If the input file contains \[ \end{proof} \] then $\LaTeX$ finishes off the displayed maths and gets ready for a new line before it reads any instructions connected with ending the proof. But traditionally the end-of-proof sign goes in the display, not on a new line. So you just have to put it in by hand in every proof.

64 Symbols for the number sets

It is a good idea to have commands such as $\mathbb{R}$ for the real numbers and other standard number sets. Traditionally these were typeset in bold. Because mathematicians usually do not have access to bold chalk, they invented the special symbols that are now often used for $\mathbb{R}$, $\mathbb{C}$, etc. These symbols are known as “blackboard bold”. Before insisting on using them, consider whether going back to the old system of ordinary bold might not be acceptable (it is certainly simpler).

A set of blackboard bold capitals is available in the AMS fonts “msam” (e.g., “msam10” for 10pt) and “msbm”. The fonts have a large number of mathematical symbols to supplement the ones in the standard $\TeX$ distribution. The fonts are available in fonts/ams/amsfonts/sources/symbols

Two files which load the fonts and define the symbols are provided, and both work with either $\TeX$ or $\LaTeX$. Questions or suggestions regarding these fonts should be directed to tech-support@math.ams.org.

Another complete set of blackboard bold fonts, the bbob family, is available in METAFONT (in fonts/bbold). This set has the interesting property of offering blackboard bold forms of lower-case letters, something rather rarely seen on actual blackboards.

The “lazy person’s” blackboard bold macros:

```latex
\newcommand{\R}{\sf {R\hspace{-0.9ex}\rule{0.15ex}{1.5ex}\hspace*{0.9ex}}}
\newcommand{\N}{\sf {N\hspace{-1.0ex}\rule{0.15ex}{1.3ex}\hspace*{1.0ex}}}
\newcommand{\Q}{\sf {Q\hspace{-1.1ex}\rule{0.15ex}{1.5ex}\hspace*{1.1ex}}}
\newcommand{\C}{\sf {C\hspace{-0.9ex}\rule{0.15ex}{1.3ex}\hspace*{0.9ex}}}
```

work well at normal size if the surrounding text is cmr10. However, they are not part of a proper maths font, and so do not work in sub- and superscripts. Moreover, the size and position of the vertical bar is affected by the font of the surrounding text.
Roman theorems

If you want to take advantage of the powerful `\newtheorem` command without the constraint that the contents of the theorem is in a sloped font (for example, to use it to create remarks, examples, proofs, ...) then you can use the style file `theorem.sty` (part of `macros/latex/packages/tools`). Alternatively, the following sets up an environment `remark` whose content is in roman.

\newtheorem{preremark}{Remark}
\newenvironment{remark}{\begin{preremark}\rm}{\end{preremark}}

This will not work if you are using NFSS (see question 94) outside of `LaTeX2ε` (see question 95), because the command `\rm` behaves differently there.

Labels on lists

If you want your top-level `enumerate`s to be labelled ‘I’, ‘II’, ..., then give these commands:

\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\labelenumi}{\theenumi/}

The possible styles of numbering are given in Section 6.3 of Lamport’s book (see question 17). If you are lazy and just change `\labelenumi` in one go then your cross-references will be wrong.

For lower level enumerates, replace `enumi` by `enumii`, `enumiii` or `enumiv`, according to the level. If your label is much larger than the default, you should also change `\leftmargini`, `\leftmarginii`, etc.

If you’re running `LaTeX2ε`, the package `enumerate.sty` (part of `macros/latex/packages/tools`) offers similar facilities. With `enumerate.sty`, the example above would be achieved simply by starting the enumeration `\begin{enumerate}\[I/\].

Unnumbered sections in the Table of Contents

The easiest way to get headings of funny ‘sections’ such as prefaces in the table of contents is to use the counter `secnumdepth` described in Appendix C of the `LaTeX` manual. For example:

\setcounter{secnumdepth}{-1}
\chapter{Preface}

Of course, you have to set `secnumdepth` back to its usual value (which is 2 in the standard styles) before you do any ‘section’ which you want to be numbered.

This is why it works. `\chapter` without the star does

1. put something in the `.toc` file;
2. if `secnumdepth ≥ 0`, increase the counter for the chapter and write it out.
3. write the chapter title.

Other sectioning commands are similar, but with other values used in the test.

Style of section headings

Suppose that the editor of your favourite journal has specified that section headings must be centred, in small capitals, and subsection headings ragged right in italic, but that you don’t want to get involved in the sort of programming described in `The \LaTeX\ Companion` (see question 17; the programming itself is discussed in question 85). The following hack will probably satisfy your editor. Define yourself new commands

\newcommand{\ssection}[1]{%\section[\#1]{\protect\centering\sc #1}}
\newcommand{\ssubsection}[1]{%\subsection[\#1]{\protect\raggedright\it #1}}

and then use `\ssection` and `\ssubsection` in place of `\section` and `\subsection`. This isn’t perfect: section numbers remain in bold, and starred forms need a separate redefinition. Also, this will not work if you are using NFSS (see question 94) outside of `LaTeX2ε` (see question 95), because the font-changing commands behave differently there.

Indent after section headings

`LaTeX` implements a style that doesn’t indent the first paragraph after a section heading. There are coherent reasons for this, but not everyone likes it. The package `indentfirst.sty` (part of `macros/latex/packages/tools`) suppresses the mechanism, so that the first paragraph is indented.
70  Changing the margins in \LaTeX
Don’t do it. Learn some \LaTeX, produce some documents, and then ask again.

You can never change the margins of a document by software, because these depend on the actual size of the paper, which may be different from what your software thinks. (For example, standard US paper is a different width from A4 paper.) What you can change are the distances from the apparent top and left edges of the paper, and the width and height of the text. Changing the last two requires more skill than you might expect. The height should bear a certain relationship to \baselineskip. And the width should not be more than 75 characters. Lamport’s warning in his section on ‘Customizing the Style’ really must be taken seriously. One-inch margins on A4 paper are fine for 10- or 12-pitch typewriters, but not for 10pt type (or even 11pt or 12pt) because so many characters per line will irritate the reader.

71  Insisting on changing the margins in \LaTeX
This answer first helps you change the margins throughout a document, then tells you how to change the margins in a portion of the document.

Perhaps the easiest way to get more out of a page in \LaTeX is to get macros/latex209/contrib/misc/fullpage.sty, which sets the margins of the page identical to those of plain \TeX, i.e., 1-inch margins at all four sides of the paper. It also contains an adjustment for A4 paper.

Somewhat more flexible is macros/latex/contrib/other/misc/vmargin.sty, which has a canned set of paper sizes (a superset of that provided in \LaTeX2\epsilon), provision for custom paper, margin adjustments and provision for two-sided printing.

Here is a brief explanation of \LaTeX’s page parameters. For details, see section C.5.3 of the \LaTeX manual (pp. 181–182). The origin in \texttt{dvi} coordinates is one inch from the top of the paper and one inch from the left side. This explains the “one inch less than” terminology used in the \LaTeX manual. In \texttt{dvi} coordinates, positive horizontal measurements extend right across the page, and positive vertical measurements extend down the page. Thus, for margins closer to the left and top edges of the page than 1 inch, the corresponding parameters, i.e., \texttt{\oddsidemargin}, \texttt{\evensidemargin}, \texttt{\topmargin}, \texttt{\parskip}, can be set to negative values.

To change the margins of a document within the document, modifying the parameters shown in Figure C.3 will not work. They can only be changed in the preamble of the document, i.e., before the \begin{document} statement. To adjust the margins within a document we define an environment:

\begin{verbatim}
\newenvironment{changemargin}[2]{{}
  \begin{list}{}{%
    \setlength{\topsep}{0pt}%
    \setlength{\leftmargin}{#1}%
    \setlength{\rightmargin}{#2}%
    \setlength{\listparindent}{\parindent}%
    \setlength{\itemindent}{\parindent}%
    \setlength{\parskip}{0pt plus 1pt}%
  }%
  \item[]\end{list}}%
\end{verbatim}

This environment takes two arguments, and will indent the left and right margins by their values, respectively. Negative values will cause the margins to be narrowed, so \texttt{\begin{changemargin}{-1cm}{-1cm}} narrows the left and right margins by 1cm. (Note that the value given for \texttt{\parskip} is the default for \LaTeX; what should be there is “the value in the surrounding text”.)

72  Finding the width of a letter, word, or phrase
Put the word in a box, and measure the width of the box. For example,

\begin{verbatim}
\newdimen\stringwidth
\setbox0=\hbox{hi}
\stringwidth=\wd0
\end{verbatim}

Note that if the quantity in the \hbox is a phrase, the actual measurement only approximates the width that the phrase will occupy in running text, since the inter-word glue can be adjusted in paragraph mode.

The same sort of thing is expressed in \LaTeX by:

\begin{verbatim}
\newlength{\gnat}
\settowidth{\gnat}{\textbf{small}}
\end{verbatim}

This sets the value of the length command \texttt{\gnat} to the width of “small” in emboldened text.
Excluding blocks of text from the .dvi file

Rainer Schöpf's verb.sty provides a comment environment which excludes everything between \begin{comment} and \end{comment}. This package is available as part of macros/latex/packages/tools.

A more general environment for doing the job is Victor Eijkhout's comment.sty, which lets you define environments for inclusion or exclusion in a document, thus offering a primitive configuration structure. It is available from the CTAN sites in macros/latex209/contrib/misc/comment.sty.

Defining a new log-like function in \TeX

Use the \texttt{\textbackslash mathop} command, as in:

\begin{verbatim}
\newcommand{\diag}{\mathop{\text{diag}}}
\end{verbatim}

Subscripts and superscripts on \texttt{\textbackslash diag} will be placed exactly as they are on \texttt{\textbackslash lim}. If you want your subscripts and superscripts always placed to the right, do:

\begin{verbatim}
\newcommand{\diag}{\mathop{\text{diag}}\nolimits}
\end{verbatim}

This works in \TeX{} 2.09 and in \TeX{} 2ε, but not under NFSS alone (see question [81]). There is an improved method for doing this in \TeX{} 2ε in the works; in the interim, one can use the the \texttt{\textbackslash operatorname} command of \texttt{amstex.sty} (which is part of the \texttt{AMS-\TeX} package: fonts/ams/amslatex).

(It should be noted that “log-like” was reportedly a joke on Lamport's part; it is of course clear what was meant.)

Typesetting all those \TeX-related logos

Knuth was making a particular point about the capabilities of \TeX{} when he defined the logo. Unfortunately, many believe, he thereby opened floodgates to give the world logos such as \texttt{AM\TeX{}}, \texttt{Pic\TeX{}}, \texttt{Bib\TeX{}}, and so on. Lamport invented \TeX{} 2ε, and marketing input led to the current logo \TeX{} 2ε.

The common people don’t have to follow this stuff wherever it goes, but, for those who insist, a large collection of logos is defined in macros/eplain/texnames.sty; the METAFONT logo can be set in fonts that \TeX{} 2ε knows about (so that it scales with the surrounding text) using the package macros/latex/contrib/supported/mflogo.

For those who don’t wish to acquire the ‘proper’ logos, the canonical thing to do is to say \texttt{AMS-\TeX{}} for \texttt{AM\TeX{}}, \texttt{Pic\TeX{}} for \texttt{Pic\TeX{}}, \texttt{Bib\TeX{}} for \texttt{Bib\TeX{}}, and so on.

Things are Going Wrong...

Weird hyphenation of words

You may have a version mismatch problem. The hyphenation system changed between version 2.9 and 3.0. If you are using (plain)\TeX{} version 3.0 or later, make sure your \texttt{plain.tex} file has a version number which is at least 3.0; if you are using \TeX{} 2.09 (the last version of which was released on 25 March 1992), you need to upgrade to \TeX{} 2ε, since the sources of \TeX{} 2.09 are no longer publicly available.

If you’re using \TeX{} 2ε, the problem probably arises from your \texttt{lthyphen.cfg} file, which has to be created if you’re using a multi-lingual version.

For the curious, here’s what happened: before \TeX{} 3.0 the hyphenation algorithm would not break a word if the part before the break was not at least two characters long, and the part after the break at least three characters long. Starting with version 3.0 the parameters \texttt{\lefthyphenmin} and \texttt{\righthyphenmin} control the length of these fragments. These are set to 2 and 3, respectively, in the new \texttt{plain} and \texttt{lplain} formats. They can be set to any value, of course, but if \texttt{\lefthyphenmin} + \texttt{\righthyphenmin} is greater than 62, all hyphenation is suppressed.

(Grupo) peculiar hyphenation

You may have found that \TeX{}’s famed automatic word-division does not produce the break-points recommended by your dictionary. This may be because \TeX{} is set up for American English, whose rules for word division (as specified, for example, in Webster’s Dictionary) are completely different from the British ones (as specified, for example, in the Oxford Dictionaries). This problem is being addressed by the UK \TeX{} User community (see Baskerville, issue 4.4) but an entirely satisfactory solution will take time. An interim hyphenation file is available in language/english/ukhyph.tex.

Enlarging \TeX

People sometimes get messages saying ‘memory capacity exceeded’. Most of the time this error can be fixed \textit{without} enlarging \TeX. The most common causes are unmatched braces, extra-long lines, and poorly-written macros. Extra-long lines are often introduced when files are transferred incorrectly between operating systems, and line-endings are
not preserved properly (the tell-tale sign of an extra-long line error is when the complaint is that the ‘buf_size’ has overflowed).

If you really need to extend your \TeX’s capacity, the proper method depends on your installation. In the purest form, you change the parameters in module 11 of the WEB source. In less pure forms, you might need to modify a change file, or perhaps change some environment variables; em\TeX allows you to adjust the memory allocation criteria on the command line. Consult the documentation that came with your implementation.

79 Moving tables and figures in \LaTeX

Tables and figures have a tendency to surprise, by floating away from where they were specified to appear. This is in fact perfectly ordinary document design; any professional typesetting package will float figures and tables to where they’ll fit without violating the certain typographic rules. Even if you use the placement specifier \texttt{h} for ‘here’, the figure or table will not be printed ‘here’ if doing so would break the rules; the rules themselves are pretty simple, and are given on page 198, section C.9 of the \LaTeX manual. In the worst case, \LaTeX’s rules can cause the floating items to pile up to the extent that you get an error message saying “Too many unprocessed floats”; this means that the limited set of registers in which \LaTeX stores floating items is full. What follows is a simple checklist of things to do to solve these problems (the checklist talks throughout about figures, but applies equally well to tables).

- Are the placement parameters on your figures right? The default (tbp) is reasonable; you should never simply say \texttt{h}, for example, since that says “if it can’t go here, it can’t go anywhere”, and as a result all subsequent floats pile up behind it.
- Can you perhaps prevent your figures from floating by adjusting \LaTeX’s placement parameters? Again, the defaults are reasonable, but can be overridden in case of problems. The parameters are described on pages 199–200, section C.9 of the \LaTeX manual.
- Are there places in your document where you could ‘naturally’ put a \texttt{\clearpage} command? If so, do: the backlog of floats is cleared after a \texttt{\clearpage} command. (Note that the \texttt{\chapter} command implicitly executes \texttt{\clearpage}, so you can’t float past the end of a chapter.)
- Have a look at the \LaTeX 2\epsilon afterpage package (part of macros/latex/packages/tools). Its documentation gives as an example the idea of putting \texttt{\clearpage after} the current page (where it will clear the backlog, but not cause an ugly gap in your text), but also admits that the package is somewhat fragile.
- As a last resort, try the package macros/latex209/contrib/misc/morefloats.sty; this ‘simply’ increases the number of floating inserts that \LaTeX can handle at one time (from 18 to 36), but that may suit your needs.
- If you actually wanted all your figures to float to the end (e.g., for submitting a draft copy of a paper), don’t rely on \LaTeX to do it for you: get the endfloat package to do the job for you (macros/latex/contrib/supported/endfloat).

80 \texttt{\pagestyle{empty}} on first page in \LaTeX

If you use \texttt{\pagestyle{empty}}, but the first page is numbered anyway, you are probably using the \texttt{\maketitle} command too. This is not a bug but a feature! The standard \LaTeX styles are written so that initial pages (pages containing a \texttt{\maketitle}, \texttt{\part}, or \texttt{\chapter}) have a different page style from the rest of the document. Hence, the commands internally issue \texttt{\thispagestyle{plain}}. This is usually not acceptable behaviour if the surrounding page style is ‘empty’.

Possible workarounds include:

- Put \texttt{\thispagestyle{empty}} immediately after the \texttt{\maketitle} command, with no blank line between them.
- Use fancyheadings.sty, which allows you to customise the style for initial pages independently of that for body pages. It is available in macros/latex/contrib/other/fancyheadings

81 Odd behaviour of \texttt{\rm}, \texttt{\bf}, etc.

If commands such as \texttt{\rm} and \texttt{\bf} have suddenly stopped working in \LaTeX in the way that you expect, it is likely that your system administrator has installed a version of \LaTeX 2.09 with NFSS (see question 94). Complain loudly; ask your system administrator to replace this version with \LaTeX 2\epsilon (see question 95), in which commands such as \texttt{\rm} and \texttt{\bf} work just as before if you are using one of the standard classes—article, report and book (among others). In the meantime, use the option oldlfont.sty, which should have been installed at the same time as NFSS.
82 Old \texttt{\LaTeX} font references such as \texttt{\tenrm}

\LaTeX\ 2.09 defined a large set of commands for access to the fonts that it had built in to itself. For example, various flavours of \texttt{cmr} could be found as \texttt{\fivrm}, \texttt{\sixrm}, \texttt{\sevrm}, \texttt{\egtrm}, \texttt{\ninrm}, \texttt{\tenrm}, \texttt{\elvrm}, \texttt{\twlrm}, \texttt{\frtnrm}, \texttt{\svtnrm}, \texttt{\twtyrm} and \texttt{\twfvrm}.

These commands were never documented, but certain packages nevertheless used them to achieve effects they needed.

Since the commands weren’t public, they weren’t included in \LaTeX\ 2\epsilon; to use the unconverted \LaTeX\ 2.09 packages under \LaTeX\ 2\epsilon, you need also to include the package \texttt{rawfonts.sty} (which is part of the \LaTeX\ 2\epsilon distribution).

83 Missing symbols

If some symbols, such as \texttt{\Box} and \texttt{\lhd}, no longer appear to exist, then your system administrator has probably upgraded your version of \LaTeX\ to either NFSS (see question 94) or \LaTeX\ 2\epsilon (see question 95). In the former case, use \texttt{oldlfont.sty}, as in the question 81. In the latter, use the package \texttt{latexsym}, which is part of the standard \LaTeX\ 2\epsilon distribution, or the package \texttt{amsfonts}, if it is available.

84 \LaTeX\ gets cross-references wrong

Sometimes, however many times you run \LaTeX, the cross-references are just wrong. Remember that the \texttt{\label} command must come \texttt{after} the \texttt{\caption} command, or be part of it. For example,

\begin{verbatim}
\begin{figure}
\caption{A Figure}
\label{fig}
\end{figure}
\end{verbatim}

85 \texttt{@} and \texttt{\@} in macro names

A common source of problems in a \LaTeX\ document is the diagnostic about the appearance of the command \texttt{@}, or about other commands containing the character \texttt{\@}. The most common complaint is “You can’t use ‘\texttt{\spacefactor}’ in vertical mode”, but other similar ones occur.

Such problems are usually caused by including a \LaTeX\ 2\epsilon class or package file into a \LaTeX\ document by some means other than \texttt{\documentclass} or \texttt{\usepackage}. \LaTeX\ defines internal commands whose names contain the character \texttt{\@}; this enables it to avoid clashes between its internal names and names that we would normally use in our documents. In order that these commands may work at all, \texttt{\documentclass} and \texttt{\usepackage} play around with the meaning of \texttt{\@}. Solve this problem by using the correct command to include the file.

But, you will say, “\textit{The \LaTeX\ Companion tells me to use commands containing \texttt{\@}!”

Indeed; for example, there’s a lengthy section about \texttt{\@startsection} and how to use it to control the appearance of section titles. Page 15 of \textit{The Companion} explains this; and suggests that you make such changes in the document preamble, between \texttt{\makeatletter} and \texttt{\makeatother}. So the definition of \texttt{\subsection} on page 26 could be:

\begin{verbatim}
\makeatletter
\renewcommand{\subsection}{\@startsection
{subsection}{\@}{\@}{\normalsize}{\itshape}{\@}}
\makeatother
\end{verbatim}

86 Where are the \texttt{msx} and \texttt{msy} fonts?

The \texttt{msx} and \texttt{msy} fonts were designed by the American Mathematical Society in the very early days of \TeX, for use in typesetting papers for mathematical journals. They were designed using the ‘old’ \texttt{METAFONT}, which wasn’t portable and is no longer available; for a long time they were only available in 300dpi versions which only imperfectly matched modern printers. The AMS has now redesigned the fonts, using the current version of \texttt{METAFONT}, and the new versions are called the \texttt{msa} and \texttt{msb} families; they are available from \texttt{fonts/ams/amsfonts/sources/symbols}

Nevertheless, \texttt{msx} and \texttt{msy} continue to turn up to plague us. There are, of course, still sites that haven’t got around to upgrading; but, even if everyone upgraded, there would still be the problem of old documents that specify them.

If you have a \texttt{.tex} source that requests \texttt{msx} and \texttt{msy}, the best technique is to edit it so that it requests \texttt{msa} and \texttt{msb} (you only need to change the single letter in the font names).
If you have a \texttt{dvi} file that requests the fonts, there is a package of virtual fonts (see question \ref{q:vf-files}) to map the old to the new series; it’s available in \texttt{fonts/vf-files/msx2msa}

\section*{Where are the \texttt{am} fonts?}
One still occasionally comes across a request for the \texttt{am} series of fonts. The initials stood for ‘Almost [Computer] Modern’, and they were the predecessors of the Computer Modern fonts that we all know and love (or hate\footnote{The fonts acquired their label ‘Almost’ following the realisation that their first implementation in \texttt{METAFONT79} still wasn’t quite right; Knuth’s original intention had been that they were the final answer}). There’s not a lot one can do with these fonts; they are (as their name implies) almost (but not quite) the same as the \texttt{cm} series; if you’re faced with a document that requests them, all you can reasonably do is to edit the document. The appearance of \texttt{dvi} files that request them is sufficiently rare that no-one has undertaken the mammoth task of creating a translation of them by means of virtual fonts; however, most drivers let you have a configuration file in which you can specify font substitutions. If you specify that every \texttt{am} font should be replaced by its corresponding \texttt{cm} font, the output should be almost correct.

\section*{‘String too long’ in \texttt{BIBTEX}}
The \texttt{BIBTEX} diagnostic “Warning—you’ve exceeded 1000, the \texttt{global-string-size}, for entry foo” is not one that you can hope to avoid by altering the \texttt{BIBTEX} style in a simple way — \texttt{BIBTEX} itself needs recompiling to increase its limit on string sizes (which is often not practical, and is never desirable). You must therefore address the problem by changing your bibliography database.

The problem usually arises from a very large abstract or annotation included in the database. The only way forward is to take the entry out of the database, so that you don’t encounter \texttt{BIBTEX}’s limit, but you may need to retain the entry because it will be included in the typeset. In such cases, put the body of the entry in a separate file:

\begin{verbatim}
@article{long.boring, 
  author = "Fred Verbose",
  abstract = "{\input{abstracts/long.tex}}"
}
\end{verbatim}

In this way, you arrange that all \texttt{BIBTEX} has to deal with is the file name, though it will tell \texttt{TeX} (when appropriate) to include all the long text.

\section*{Why does it do that?}

\subsection*{Why does it ignore paragraph parameters?}
When \TeX{} is laying out text, it doesn’t work from word to word, or from line to line; the smallest complete unit it formats is the paragraph. The paragraph is laid down in a buffer, as it appears, and isn’t touched further until the end-paragraph marker is processed. It’s at this point that the paragraph parameters have effect; and it’s because of this sequence that one often makes mistakes that lead to the paragraph parameters not doing what one would have hoped (or expected).

Consider the following sequence of \LaTeX: 

\begin{verbatim}
\raggedright % declaration for ragged text 
Here’s text to be ranged left in our output, 
but it’s the only such paragraph, so we now 
end the group.}
\end{verbatim}

\begin{verbatim}
Here’s more that needn’t be ragged... 
\end{verbatim}

\TeX{} will open a group, and set the ragged-setting parameters within that group; it will then save a couple of sentences of text and close the group (thus restoring the previous value of the ragged-setting parameters). Then it encounters a blank line, which it knows to treat as a \texttt{\par} token, so it typesets the two sentences; but because the enclosing group has now been closed, the parameter settings have been lost, and the paragraph will be typeset normally.

The solution is simple: close the paragraph inside the group, so that the setting parameters remain in place. An appropriate way of doing that is to replace the last three lines above with:

\begin{verbatim}
end the group.\par
Here’s more that needn’t be ragged... 
\end{verbatim}
In this way, the paragraph is completed while the setting parameters are still in force within the enclosing group.

90 What's the reason for 'protection'?

Sometimes \LaTeX{} saves data it will reread later. These data are often the argument of some command; they are the so-called moving arguments. (Moving' because data are moved around.) Places to look for are all arguments that may go into table of contents, list of figures, etc.; namely, data that are written to an auxiliary file and read in later. Other places are those data that might appear in head- or footlines. Section headers and figure captions are the most prominent examples; there’s a complete list in Lamport’s book (see question 17).

What’s going on really, behind the scenes? The commands in the moving arguments are already expanded to their internal structure during the process of saving. Sometimes this expansion results in invalid \TeX{} code when processed again. “\protect\cmd{}” tells \LaTeX{} to save \cmd{} as \cmd{}, without expansion.

What is a ‘fragile command’? It’s a command that expands into illegal \TeX{} code during the save process.

What is a ‘robust command’? It’s a command that expands into legal \TeX{} code during the save process.

No-one (of course) likes this situation; the \LaTeX{}3 team have removed the need for protection of some things in the production of \LaTeX{}2ε, but the techniques available to them within current \LaTeX{} mean that this is an expensive exercise. It remains a long-term aim of the team to remove all need for these things.

91 Why doesn’t \texttt{verb} work within…?

The \LaTeX{} verbatim commands work by changing category codes. Knuth says of this sort of thing “Some care is needed to get the timing right…”, since once the category code has been assigned to a character, it doesn’t change. So \verb{} has to assume that it is getting the first look at its parameter text; if it isn’t, \TeX{} has already assigned category codes so that \verb{} doesn’t have a chance. For example:

\begin{verbatim}
\verb+error+
\end{verbatim}

will work (typesetting ‘\error{}’), but

\begin{verbatim}
def\unbrace{\verb+error+}{#1}
\unbrace{\verb+error+}
\end{verbatim}

will not (it will attempt to execute \error{}).

This is why the \LaTeX{} book insists that verbatim commands must not appear in the argument of any other command; they aren’t just fragile, they’re quite unusable in any command parameter, regardless of \protect{} (see question 90).

92 Case-changing oddities

\TeX{} provides two primitive commands \texttt{\uppercase} and \texttt{\lowercase} to change the case of text; they’re not much used, but are capable creating confusion.

The two commands do not expand the text that is their parameter — the result of \texttt{\uppercase{abc}} is ‘ABC’, but \texttt{\uppercase{\abc{}}} is always ‘\abc{}`, whatever the meaning of \abc{}. The commands are simply interpreting a table of equivalences between upper- and lowercase characters. They have (for example) no mathematical sense, and

\begin{verbatim}
\uppercase{About $y=f(x)$}
\end{verbatim}

will produce

ABOUT $Y=F(X)$

which is probably not what is wanted.

The same sort of problem can arise with \LaTeX{} environment names, on occasions when \LaTeX{} employs \texttt{\uppercase} implicitly. For example, the standard classes use \texttt{\uppercase} to set the marks according to the arguments of \texttt{\chapter} and \texttt{\section} commands, so that a chapter whose title contains a small table expressed in terms of \texttt{\begin{tabular}… \end{tabular}}, will set a mark containing \texttt{\begin{TABULAR}, etc., which is an environment that doesn’t exist.

93 Why are # signs doubled in macros?

The way to think of this is that ## gets replaced by # in just the same way that #1 gets replaced by ‘whatever is the first argument’.

So if you define a macro and use it as:

\begin{verbatim}
\def\a#1{...#1...#1...#1...} \a{b}
\end{verbatim}

the macro expansion produces ‘…b…b…b…’, which people find normal. However, if we now fill in the ‘…’:

\begin{verbatim}
\def\a{----#1----}\def\x #1{xxx#1}
\end{verbatim}

–34–
\a(b) will expand to ‘---b---\def\x b(\x x b)’.
This defines \x to be a macro delimited by \b, and taking no arguments, which people may find strange, even though it is just a specialisation of the example above. If you want \a to define \x to be a macro with one argument, you need to write:
\def\a#1{---#1---\def\x ##1{xxx##1}}
and \a(b) will expand to ‘---b---\def\x #1{xxx#1}’, because #1 gets replaced by ‘b’ and ## gets replaced by #.

To nest a definition inside a definition inside a definition then you need ####1, as at each stage ## is replaced by #. At the next level you need 8 #s each time, and so on.

Q Recent Developments

94 The New Font Selection Scheme (NFSS)
NFSS was an extension to \LaTeX\ written by Frank Mittelbach and Rainer Schöpf. It is described in TUGboat, 1989 10(2). In traditional typesetting, fonts are described by four parameters: the family (e.g., computer modern), the series (i.e., the weight and width of the font, such as light or bold), the shape (e.g., italic), and the size. NFSS is a mechanism allowing the user to change any of these independently. NFSS makes it relatively easy to use nonstandard fonts such as the PostScript ones with \LaTeX, and easy to change maths fonts. It also allows dynamic loading of fonts at runtime (i.e., not when the format file is created).

With the demise of \LaTeX\ 2.09 as supported software, the label ‘NFSS’ has become somewhat misleading, as there’s no ‘old’ scheme with which to contrast it — \LaTeX\ has incorporated the NFSS.

95 \LaTeX\ 2e (the new standard \LaTeX\)
\LaTeX\ 2e is a new version of the \LaTeX\ package, prepared and supported by the \LaTeX\3 project team. It moved out of its test phase in June 1994, and is now the standard \LaTeX; \LaTeX\ 2.09 is no longer supported.
\LaTeX\ 2e is upwardly compatible with \LaTeX\ 2.09, but has new features, including:
- NFSS (see question [94]) is now standard.
- SL\LaTeX\ is consolidated into it, so that there is no longer a need for a separate format.
- Better control of floating environments, such as figures.
- There is a documented interface for package and class writers (though not yet for designers).
- Enhanced box commands, e.g., options to specify the height of a minipage.
- \ref can be used in \caption without being protected.
- \newcommand can define commands with one optional argument.
- A standard package for colour and graphics inclusion.

Since \LaTeX\ 2e is supported, you can report bugs or problems with it by typing ‘latex latexbug’ and sending the report thus generated to latex-bugs@uni-mainz.de.

96 The \LaTeX\3 project
The \LaTeX\3 project team is a small group of volunteers whose aim is to produce a major new document processing system based on the principles pioneered by Leslie Lamport in the current \LaTeX. It will remain freely available and it will be fully documented at all levels.

The \LaTeX\3 team has already delivered its first product, \LaTeX\3 (see question [95], a macro package based on Lamport’s original code, but modified to be more readily supportable than was Lamport’s.

97 Using \TeX\ to typeset from SGML files
Jonathan Fine (J.Fine@pmms.cam.ac.uk) is developing software which will allow \TeX\ to typeset directly from SGML files. The work is producing a macro package that directly interprets an SGML source file, as opposed to the conversion programs outlined earlier (see question [45]).

The work is beginning to show fruits of success, and an announcement will be made early in 1995.

98 The Omega project
Omega (Ω) is a program built on top of \TeX\ which works internally with 16-bit characters (Unicode); this allows it to work with most scripts in the world without any complications of coding schemes. Omega also has a powerful concept of input and output filters to allow the user to work with existing transliteration schemes, etc. Omega is an ongoing project by John Plaice (plaice@ift.ula.val.ca) and Yannis Haralambous (Yannis.Haralambous@univ-lille1.fr). Contact Yannis to get on the discussion list.

–35–