

what they could achieve with a simple text editor and a L^AT_EX previewer. In terms of the abilities of Scientific *Word*, I have no answer to this. If you are comfortable composing L^AT_EX with a text editor, then I would not try to persuade you otherwise. Scientific *Word* is software for the occasional L^AT_EX user rather than the expert. Scientific *WorkPlace*, however, is a different matter. This is a working environment for anyone involved in the mathematical sciences, in which problems can be written in as near to standard mathematical language as one is going to get, and solutions to those problems derived using the numerical and symbolic calculation facilities of Maple. In this respect its nearest competitor is MathCad, which also uses Maple as its calculating engine, but Scientific *WorkPlace* has the one big advantage that it uses L^AT_EX as its native file format. Also, as well as including Maple, Scientific *WorkPlace* offers the ability to link to Mathematica, if this is already installed on your system. If, like me, you admire the Mathematica kernel, but find its notebook front-end awkward and cumbersome to use then Scientific *WorkPlace* will provide a welcome alternative.

Do not expect to be able to view and edit absolutely any L^AT_EX file with Scientific *WorkPlace*. There are occasions when documents cannot be loaded because “SW cannot handle certain `\newcommand` or `\def` statements in the L^AT_EX preamble”. However, having said this, after choosing a couple of files at random from the Los Alamos preprint archive, I had no trouble loading them into Scientific *WorkPlace*.

In the current version of the software, a distinction has been drawn between the *style* of the document, which controls the on-screen appearance and the non-typeset printing, and the L^AT_EX typesetting specifications for the document. Earlier versions of the product used the terminology ‘*style*’ to refer to the typesetting specifications, a usage which has confusingly remained in the title of the program that allows new typesetting specifications to be created: the *style editor*. If like me, you have always been daunted by the prospect of creating your own `.sty` files, this style editor could be a useful tool. The time available for completing this review was too short to be able to offer a fully considered opinion on its abilities, but in a relatively short time I *was* able to create a reasonable `.sty` file, something I have never before attempted in five years of L^AT_EX use. For many purposes, however, there is no need to even attempt to create your own typesetting specifications. The built-in document shells include specific-

ations for many well-known journals, such as *Monthly Notices of the Royal Astronomical Society* and *IEEE Transactions*. With version 1.0 of Scientific *Word*, there was a very useful booklet which consisted of print-outs of sample pages from all the then-supported print styles. I would have liked to see something similar with the present version. It is also difficult to know how one should proceed to set up Scientific *WorkPlace* to work with styles for journals not included in the built-in shells. I have just downloaded `chsm.cls` from the *Chicken and Sheep Monthly* website. How should I now proceed to be able to work with this style within Scientific *WorkPlace*? I suspect it involves something to do with a rather frightening button labelled ‘go native’, but I am not sure of this, and the program documentation needs to give much clearer instructions in these circumstances.

It seems obligatory these days for any major piece of software to offer the ability to ‘publish documents onto the Web and let others view, download and work with your documents’. Scientific *WorkPlace* is no exception. I am unconvinced that this is a great enhancement; L^AT_EX, as a typesetting medium, fits uncomfortably with the style of the Web, which in this context is best regarded as a way of delivering documents that can be viewed and processed off-line. The true future of mathematical documents on the Web must be via MathML, and I very much hope that MacKichan are working on incorporating an ability to save as MathML into their equation editor.

As a final point, earlier versions of this software produced a L^AT_EX output that was not portable, in the sense that when sending the file to someone working with a standard L^AT_EX installation they would complain that there was a missing file. Now there is an option to save as portable L^AT_EX, which does not insert the line `\input{tclatex}` in the `.tex` file for the document or include additional L^AT_EX macros in the `.tex` file. I will be using this option to produce an output file to send this review to the editor of *Baskerville*, so if there is no Editor’s Note of complaint attached, you can assume that it worked!⁹

It is planned that there will be a brochure together with a 30-day demo CD going out with each copy of the issue of *Baskerville* in which this review appears. I strongly recommend that you try it for yourself.

Gareth Suggett

Email: gsuggett@wiley.co.uk



Historical snippets on T_EX and friends

During the early development of the T_EX and METAFONT programs, Don Knuth distributed a set of outline fonts under the name ‘Almost Modern’ (`amr10`, etc.). These fonts were soon considered to be a preliminary trial release.¹⁰ Knuth later made extensive changes and refinements to the META-

FONT programs and to the appearance of the fonts, and they were finally released under the name ‘Computer Modern’. The old ‘Almost Modern’ continued to appear for some years in isolated sites, although their use was strongly discouraged. They were easily recognised—in contrast to Computer

⁹The source was indeed reasonably portable. – Ed.

¹⁰With both T_EX and METAFONT, Knuth demonstrated the precept “plan to throw one away”, made famous by Frederick P. Brooks Jr. in his classic *The Mythical Man-Month: Essays on Software Engineering* (Reading, Mass.: Addison-Wesley, 1975), ch. 11.

Modern—by their smaller counters and heavier, cramped appearance. Some releases of Almost Modern had a tell-tale signature: a small horizontal stroke protruding from the right notch of lower case ‘y’, a bit like this: ‘y’. The changes made to Almost Modern to produce Computer Modern were largely the result of interaction between Knuth and the type designers Hermann Zapf, Matthew Carter, Richard Southall (who did the Computer Modern sans serif), Neenie Billawalla (who did the *CALLOGRAPHIC CAPS*), Chuck Bigelow and Chris Holmes.

There was another wrinkle, to do with PK files, and to understand it we need to look back a little further into T_EX’s history. The very first versions of T_EX and METAFONT were rather more device-specific than today’s versions. For example, T_EX sometimes produced output files ready for direct ingestion by particular printers, rather as PdfT_EX today produces an alternative output format to DVI. Early METAFONT too produced pixel files in the format of particular devices. Some of these early font formats had intriguing features. For example the VNT format fonts contained each character of the font twice: once in the normal orientation, and once rotated by 90°. This was very useful for jobs such as labelling the axes of graphs. (Remember that PostScript was not yet invented, so there was no obvious way of rotating text.) Later, the ‘pixel’ or PXL format became METAFONT’s standard output. Rotated characters were abandoned, with the proviso that ‘there is a separate utility that can read a PXL file and write a rotated version into a new PXL file’.¹¹ The PXL raster description, containing uncompressed bitmaps, was easy to read, but was also very verbose, and the size of the files grew with the square of the resolution.¹² The Almost Modern fonts, for example, were normally distributed as PXL files. The PXL format had other drawbacks: one was that it could only manage fonts with up to 128 characters, which made it useless for some of the larger fonts created with METAFONT. It was fine for Computer Modern, of course.

Knuth’s decision to make METAFONT’s output device-independent led to the design of the Generic Font (GF) file

format, which was the output format of METAFONT when version 1.0 was released in 1984.¹³ The GF format is more compact than PXL was, can contain fonts of 256 characters, and has other advantages.

In 1985, however, Tom Rokicki devised the ‘packed’ or PK format which produced even better compression than the GF format.¹⁴ The GFtoPK program reduced the font files by about as much as is possible by current file compression techniques.¹⁵ Even dedicated Lempel-Ziv compression programs such as ZIP, *compress*, and ZOO cannot significantly improve the compression achieved by the PK format. Like the GF format, it allows for fonts of up to 256 characters, and it is also easier for DVI translators to process efficiently. The T_EX community settled on the PK format as its standard, and use of the PXL format faded away. Programs were made available that converted to and from each of the different font formats: PXtoPK, PKtoPX, PKtoGF, and so on.¹⁶

Why hasn’t METAFONT been altered to write PK files instead of GF? At least two reasons: GF files can contain METAFONT **specials** and **numspecials**, which may not appear in PK files.¹⁷ Such **specials** are used by GFtoDVI, for example, to print labels on proof sheets of fonts.

For a time, T_EX users could still come across PXL format fonts. For example, there was a useful program by David Strip and Dimitri Vulis, called HP2PXL, which read Hewlett-Packard soft fonts and generated a TFM that T_EX could use. It also converted the soft font bitmaps into PXL format, so that DVI drivers could set type using them.¹⁸

An ingenious use of HP2PXL was made by an anonymous person in early 1989. S/he used PC Paintbrush to write the letters of the alphabet, in handwriting, creating the file HAND.PCX. Then s/he created an HP soffont, HAND.SFP, from these handwritten letters using a program called PCX2FNT1. Finally, HP2PXL was used to convert HAND.SFP into HAND.PXL and HAND.TFM, which could be used directly in T_EX and on a 300 dpi laser printer.¹⁹ However, the wide availability of PXtoPK meant that even these PXL files were usually converted into PK format before use.

It may come as rather a shock to see T_EX setting type in handwriting! But the possibilities are intriguing.
The quick brown fox jumps over the lazy dog. 0123456789

Dominik Wujastyk



¹¹ See David Fuchs, ‘The format of PXL files’ *TUGboat* 2(3) (1981), 10^b. This utility seems to have sunk without trace.

¹² See David Fuchs, *ibidem*, 8–12, and *idem*, ‘News from the home front’, *TUGboat* 2(3) (1981), 22. There were actually two versions of the PXL file format, one (word-aligned) with ID byte 1001, the other (byte-aligned) with ID 1002, which was less verbose (*ibidem*, 10–11). ArborText, for example, distributed the latter type of file for some time.

¹³ A program by Arthur Samuel called GFtoPXL converted fonts back to the older format, for people whose DVI translators couldn’t yet read the new format.

¹⁴ See Tomas Rokicki, ‘Packed (PK) font file format’ *TUGboat* 6(3) (1985), 115–120.

¹⁵ The WEB listing for GFtoPK was published in Donald E. Knuth, Tomas G. Rokicki, and Arthur L. Samuel, *METAFONTware* (Stanford, 1989), 201–42.

¹⁶ There was an early release of the GFtoPK program which underwent improvement. Thus there were, for a short time, two kinds of PK file in circulation—one kind with ID byte 87 and another with ID 89. The latter format is the only one sanctioned today. See the GFtoPK WEB file for fuller information.

¹⁷ On METAFONT **specials**, which are not the same as T_EX **\specials**, see *The METAFONTbook*, appendix G.

¹⁸ See *TUGboat* 9(2), (1988), 149.

¹⁹ All these files and programs were available from the Channel 1 bulletin board in Boston, Massachusetts, in mid-1989.

Baskerville

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

Guest Editor: Dominik Wujastyk

Vol. 9 No. 2
August 1999

Baskerville is set in Monotype Baskerville, with Computer Modern Typewriter for literal text. Editing, production and distribution are undertaken by members of the Committee. Contributions and correspondence should be sent to baskerville@tex.ac.uk.

Editorial

The Guest Editor of the last issue of *Baskerville*, James Foster, explained in that issue how members of the UK-T_UG Committee have assumed editorial responsibility for the preparation and formatting of individual numbers of the newsletter. Like James, I am deeply grateful for, and awed by, the amount of work and expertise which Sebastian Rahtz has put into past issues of *Baskerville*. Thanks, Sebastian!

James also mentioned the hard work which Robin Fairbairns has done over the years in producing and distributing *Baskerville*. Although Robin is now liberated from these particular tasks, he is still heavily involved in supporting the T_EX community in various ways. In particular, he is the current custodian and mastermind behind the invaluable T_EX FAQ. This he maintains as a multi-format document which includes an incarnation on the world-wide web (see question 20 below).

The T_EX FAQ is a very special document, since it tackles head-on many of the most urgent and practical issues about which people need to know when they come into contact with T_EX. The actual questions which are answered in this FAQ are those which have appeared most often in the `comp.text.tex` discussion group, plus others which surface repeatedly in various fora. Although Robin is the current

maintainer of this FAQ, many people have contributed to it, as is explained in the introduction below.

The T_EX FAQ has been published in *Baskerville* twice before, in 1994 and 1995. These are the issues of *Baskerville* which I have most often lent or recommended to other T_EX users. In fact, I currently do not have the 1995 FAQ issue because I gave it away to someone who needed it as a matter of urgency! I am confident that this newly updated T_EX FAQ, now expanded to cover 126 questions, will be every bit as popular and useful as its predecessors, and will save T_EX users many hours of valuable time.

Don't overlook Gareth Suggett's review on p. 37. If you write lots of equations, this software could be for you! And finally, I wonder how many *Baskerville* readers remember the bits and pieces recalled in Historical Snippets (p. 38)? If you have never heard of any of this stuff before, then you are just the kind of vibrant, new, upcoming UK-T_UG member that we want, infinitely more interested in publishing your work than in fiddling with T_EX. But T_EX wasn't built in a day, and perhaps these Snippets give a flavour of some of the evolutionary forces which have made T_EX and friends what they are today.

Stop Press :

UK T_EX Users' Group Tenth Anniversary Meeting: Knuth to host Q&A session

The Chairman and Committee of UK-T_UG request the pleasure of your company at the Tenth Anniversary Meeting of the UK T_EX Users' Group, to be held at St Anne's College, Oxford, on Sunday 12th and Monday 13th¹ September 1999.

The meeting will commence at 11:30 on Sunday 12th September with the first of a series of talks on the theme "T_EX and its relationship to the {SG/HT/X}ML family of languages". After lunch on Sunday there will be further talks on this topic, following which there will be the 1999 Annual General Meeting of the UK T_EX Users' Group.

On Sunday evening, there will be a special Anniversary

Banquet, at which Professor Donald Knuth and his wife Jill will be our guests of honour. After the meal, Don has kindly agreed to give a spontaneous question-and-answer session, and members of UK-T_UG will be most welcome to ask Don questions on any topic which is of mutual interest.

Monday 13th will continue the theme of "T_EX and its relationship to the {SG/HT/X}ML family of languages" and may include a tutorial on XSL.

Accommodation in St Anne's will be available for those not wishing to return home between sessions.

A more detailed announcement and booking form accompany this issue of *Baskerville*.

¹The continuation to Monday is dependent on a sufficient number of knowledgeable speakers being found; final details should be known by the time this issue of *Baskerville* is distributed.