TEX in a Book Production Department

Howard Ratner
Springer-Verlag New York Inc., 175 Fifth Avenue, New York, NY 10010, USA
212-460-1655
Internet: ratner@spint.compuserve.com

Kenneth Dreyhaupt
Springer-Verlag New York Inc., 175 Fifth Avenue, New York, NY 10010, USA
212-460-1655
Internet: dreyhaupt@spint.compuserve.com

Abstract

This paper presents the point of view of a publisher's book production department when accepting author-supplied TEX manuscripts. Topics covered include tips for authors and publishers, LATEX vs. TEX vs. $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX as publishing tools, creation of house macros, and use of freelance services.

Introduction

History. The use of TEX at Springer-Verlag New York began sometime in the early 1980s with the acceptance of "camera-ready" copy from authors. Unlike our parent company in Germany, no macros were developed and no guidelines exclusive to TEX were written. Authors had only our general camera-ready guidelines to guide them and those guidelines were developed for the lowest common denominator, namely, typewriter copy.

By the middle to late 1980s, author pressure to create their books using TEX increased to the point that LATEX macros were developed using a consultant. This enabled us to do two things: accept author-supplied copy on a standardized basis and employ freelance TEX typesetters to set copy from paper manuscript. At this point output was from 300-dpi laser printers and the quality difference was clearly visible between books done in TEX and our conventional mathematics typesetters.

During this time no TEX expertise was being developed in the Book Production Department. There were two reasons for this lack of expertise: (1) at that time people who were employed in book production departments had little or no computer skills or experience; and (2) little or no equipment was available to gain that experience.

In the last two to three years the situation has changed dramatically. The tremendous increase in author use of TEX, improved output capabilities, and growing availability and experience of TEX suppliers has made TEX not only a viable part of any mathematical or physical sciences production department but a star.

Production Outlook

Costs. One of the main functions of a production department is to spend money. The success of any department is determined by not spending any more than is necessary to produce a quality product. TEX has the capability of producing a quality product at a low cost. However, several factors can negate this cost-saving possibility: (1) author fees for providing hard copy or electronic files; (2) high output costs; (3) high freelance costs for typesetting or reformatting; (4) overhead costs when authors do not use macros or have problems in the final stages of production.

Quality. The word quality can evoke different visions from authors and production editors. Both agree it means the lack of typographical errors and errors in fact. At that point, however, the visions may diverge.

The area of greatest potential conflict is design. Authors with the power of computer technology and all the fonts that a few hundred dollars can provide, want to express some creativity not only in the words that appear on the page but in the appearance of the words themselves. Book design, however, is both an art and a science. Production editors with a decade or more of experience hesitate to change department specifications to fit a particular book, but authors who have read a few dozen books want to

repeat catchy elements that they have seen in several of them. Book design and typograghy should be left to professionals. That does not mean that some authors cannot produce quality design work, but the continued submissions of "ransom note" designs should lead departments to discourage this practice.

Error prevention is an important quality consideration. Is there anything more distracting than reading a book with the appearance of a high-quality product and finding typos and other annoying errors? One service that conventional math typesetters provide is professional proofreading. This can be lost when an author or a small freelance service is setting the book. We try to recover this lost service by copyediting in the page proof stage. The copy editor performs both copyediting and proofreading functions and also serves as a design reviewer. This step has saved authors from embarrassing errors in books they thought were final and clean.

Hardware

TEX coding is the same on all platforms, but user interfaces can vary widely. The variations of interfaces cause differences in flexibility, disk management, learning curves, and of course speed. A large part of deciding on which platform to run TEX can be derived by available equipment and budget. There are high-, middle-, and low-end TEX hardware setups.

Springer uses what could be considered a "middle-end" hardware setup. We currently run TEX in a combined DOS–Macintosh environment. Our setup grew out of a basic office computing scenario. Production editors used PCs for word processing, spreadsheets, and database work. Macintoshes were recently brought in to bolster our art program and increase our desktop publishing potential. When we decided to bring desktop publishing (including TEX) into the department, we chose to upgrade our current configuration rather than start from scratch. A few more computers were added. These were equipped with more memory, faster processors, and larger storage devices. Book-length TEX files are huge. We often find these files (particularly `dvi` files and output files) in competition with other applications for storing files in our hard drives. We even added a local area network, a pair of modems, and a scanner. In this way, TEX files can be edited and coded at one station and massaged at another. Proofs can be generated on any of our bitmap and PostScript laser printers.

More important, this configuration allows us considerable flexibility in accepting author-generated files. We can take files from DOS, Macintosh, and UNIX environments, either via disk for large files or telecommunications for short ones. Authors working in the UNIX environment are asked to download their large TEX files onto DOS- or Macintosh-formatted disks. Authors working on PCs or Macintoshes send in their files as is.

DOS. TEX runs well in both the PC and Mac environments. The DOS version utilizes separate programs that can be mixed and matched. A variety of text editors, TEX engines, `dvi` previewers, and printer drivers can be used. Each can be obtained at reasonable prices. This gives great flexibility for users. An added benefit is that a user's "tried and true" text editor can be used with maximum efficiency. A major drawback is the fact that each program must be run separately in order to create hard copy. Each program requires a given amount of startup time. Adding startup times for the four necessary programs (editor, TEX engine, previewer, and print driver) easily creates a significant amount of time merely waiting for the DOS version to load.

Macintosh. The Macintosh version offers a fully integrated interface. The text editor, engine, previewer, and driver are all built-in. Unlike the DOS version, only one program is needed to create final copy. Some consider the Macintosh version easier to learn. However, the higher prices of Macintosh hardware and software make use of these machines in large quantities more difficult in a book publishing atmosphere.

Our mixed environment allows us to leave all machines application non-specific. We have not dedicated any of our machines to only running TEX. All of our machines can be used for general word processing, spreadsheets, database work, and other desktop publishing applications, as well as for TEX. The network allows us to keep work constantly moving. Text files, macros, and style files can easily be accessed throughout the network. Production editors working on a project are not anchored to only one machine, thereby increasing productivity. Some might say that TEX does not perform at its absolute best in this environment, but Springer is not in the business of becoming a composition house. As a publisher, we set out to create a working environment in which we could accept TEX files, edit last minute changes, and obtain final copy. Raw manuscript and major editing projects are routed out of house.

UNIX. An alternate, higher-end setup would utilize UNIX. Workstations and software could be purchased at a premium price. This high price tag

would allow for increased speed, true multi-tasking, and TEX working faster. UNIX systems are mainly used by programmers and as yet have made little headway into the general business marketplace.

LATEX vs. TEX vs. AMS-TEX

The three most common varieties of TEX are LATEX, plain TEX, and AMS-TEX. Each has its strengths and weaknesses.

LATEX. From a publisher's point of view, LATEX is the version of choice. It is more structured than the others. It was created by Leslie Lamport for authors to create books and other documents. The macros are very comprehensive. Authors using LATEX almost always use the well-defined macros available in LATEX, rather than creating new ones. This reliance on the LATEX coding schemes makes it very easy for publishers to apply their own macros. The publisher simply alters the pre-existing set of LATEX macros. The author need not learn any new commands. As far as the author is concerned, he is using the standard LATEX codes.

As mentioned above, the LATEX coding scheme creates a very structured design that relies more on proper layout technique than visual appeal. Good layout is often the primary ingredient for creating a good-looking book.

Plain TEX. On the other hand, plain TEX and AMS-TEX are far more flexible. Using plain TEX, the author has near total control over the look of the book. TEX starts from scratch. There are few readily available layout macros built into the system. Authors must create their own macros or define the document line by line. Many authors prefer this flexibility as it keeps them from feeling trapped in a generic design. Such an attitude can cause problems for the publisher when trying to produce the book. Few authors are trained designers, and many times what looks nice on single sheets translates poorly to a bound book.

It is possible for publishers to supply useful macros for plain TEX. As there are no initial overall layout commands in plain TEX, these macros would introduce an entirely new set of commands for the author. Therefore, authors should use these commands as they are creating their chapters. This is different from LATEX where an author is familiar with the basic LATEX commands and therefore is readily familiar with the macro commands.

A detailed set of instructions *must* accompany the macros, as the authors must be taught the macro set and how to use it. Designing plain TEX macros brings up the problems of creating macros from scratch. This topic will be detailed in the section on house macros.

Another problem can also arise from author-created plain TEX macros—a seemingly innocuous author-created macro could unknowingly conflict with the publisher's macros, making a tremendous mess when run through TEX.

AMS-TEX. AMS-TEX's greatest asset is its easy accessability to the AMS fonts. This extra set of fonts allows mathematicians to utilize a number of special mathematical characters. These fonts can be accessed by LATEX and plain TEX, but not as easily. AMS-TEX has some built-in structure. The `AMSPPT.STY` is a layout format designed for typesetting a mathematical paper. Books have different demands. Publisher macros can be used to overlay the `AMSPPT.STY`, but the coding set available is not as comprehensive as the one available for LATEX.

Using the AMS fonts can be a problem if not handled carefully. Some high-resolution output services do not have these fonts. The fonts can be obtained, but often this leads to schedule delays and font "bugs," such as character crashes and font priority problems.

In the end, plain TEX and AMS-TEX can be used to create a high-quality book, but they will require more effort on the part of the publisher and the author to do so. LATEX was designed to make books, and with a little tweaking provided by publisher-designed macros, high-quality products are ensured.

Developing House Macros

The development and use of house macros is the most critical area for success for a book production department using TEX. House macros are the meeting place for production departments and authors.

Macros should have three qualities: (1) they should be easy to use and concise; (2) they should be flexible enough to allow the author to express himself even when confined to a house style; and (3) they should be accessible. If these qualities are not there, authors will not be inclined to use the publisher's macros, and more work for the production department will result.

Ease of use depends on the accompanying documentation in the form of an instruction file as well as the generous use of comment lines within the macro itself. Flexibility is the result of the house macro operating on the format aspects of the copy only.

Macros should be available for both single-author and edited volumes.

The easiest way to get started with house macros is to hire a TEX specialist as a consultant.

There are several advantages to this: (1) the startup speed of your TEX program will be that much faster; (2) you do not use valuable in-house time on something your department is not suited for; (3) you get professional expertise and a set of macros tailored to your house style; and (4) you get follow-up down the road when the consultant works as a troubleshooter for you. The disadvantages are: (1) the cost of hiring the consultant; (2) new macros almost always have bugs that need to be worked out; (3) if you have little or no TEX knowledge and the consultant is not working closely with you, you will have a set of macros that you do not understand and that you cannot explain to your authors.

Another method for obtaining house macros is to adapt an already existing macro for your department's use. The advantages here are: (1) the cost is low because you are not paying a consultant and you are not spending a lot of in-house time; (2) the macro is generally bug-free; (3) this forces you to learn more about TEX and will enable you to adapt to unique situations when they arise.

At Springer-Verlag New York we undertook a combination of these two approaches. First, when we had no TEX expertise in-house, a consultant was hired to develop a LATEX macro for us. This enabled us to get the ball rolling on TEX and have a "welcome mat" available for our authors. We then worked to get our TEX expertise improved internally. It did not take more than a few months of accepting LATEX books for us to feel comfortable both with LATEX and with the macros that were developed for us.

We also began picking up experience with plain TEX. We borrowed a suitable plain TEX macro package from our Heidelberg production department and adapted it for our own needs. We left 95% of the macro package untouched, therefore it was free of bugs.

After two years of this kind of experience, we are now at the level where we can begin thinking of developing our own macros. We also feel comfortable helping our authors with problems on their macros.

TEX Suppliers

The use of service suppliers for TEX is a key to the success of TEX in the production environment. There are three types of services that can be provided out-of-house: keyboarding, reformatting, and full service.

Keyboarding. The availability of TEX keyboarding services has grown in the last few years. Because a lot of overhead is not needed (all that would be required would be a microcomputer, software, and printer), services that provide only keyboarding can be quite economical for math typesetting. Springer was already using keyboarding services to provide camera-ready copy. We are now moving away from that for two reasons: (1) the keyboarding service was not providing true full service; and (2) most authors for math books are already providing TEX files for our books. Our keyboarding services are now moving to the function of inserting author and copyeditor corrections to an already existing TEX file.

Reformatting. Another service necessary for out-of-house work is reformatting of existing TEX files to house specifications. Some of this work might include dimension and font changes, figure placement or spacing, running head preparation, and insertion of house macros.

Whether Production can use these services is decided by cost. A comparison must be made between the costs of this being done out-of-house, versus in-house (overhead), or just using the TEX output as a well-prepared manuscript and typesetting through non-TEX sources.

Full Service. By full service we mean accepting a paper manuscript or a TEX file on diskette and providing keyboarding, proofreading, formatting, macro writing, illustration work, and output. This is essentially providing the same services as any conventional typesetter, just using TEX to get it done. For the Production Department there is no distinction between these services and conventional typesetters. They should have to compete on a cost, quality, and schedule basis.

Fonts, Figures, and Final Output

There are currently two tracks used for outputting TEX documents: bitmap and PostScript.

Bitmap. Following the bitmap track, one primarily uses TEX's standard Computer Modern typefaces and has them output at high resolution. Non-TEX coded figures are stripped in by hand onto the final pages. This is done by the conventional method of cut-and-paste using wax, a blade, and a light table. The bitmap track provides a high-quality final product at low cost and easy maintenance, but is very limited in flexibility. Last-minute changes are difficult to make as they often mean having to restrip art.

PostScript. The PostScript track offers a much more flexible environment. A user can choose from

a huge library of PostScript fonts at varying resolutions. Figures can be placed electronically. Figures can be scanned into *encapsulated* PostScript files or drawn using various computerized art programs, then stored as an encapsulated PostScript file. These files can then be imported using TEX's `\special` command.

PostScript is the mainstream font language of the desktop publishing community. If a user creates PostScript output files rather than `dvi` files, he can take advantage of the thousands of PostScript output vendors, both locally and internationally. This huge output service community keeps the market volatile. Prices depend on resolution required, turnaround, and volume. Unfortunately, most PostScript vendors are not TEX-aware. If a user can only create `dvi` files, they have greatly reduced the number of available output services. TEX's `dvi` file concept is foreign to most service bureaus. Being able to explain the `dvi` file concept and judge a bureau's technical expertise requires more in-house knowledge on the part of the publisher.

Springer-Verlag New York is currently following the bitmap track though we are progressively moving toward the PostScript track. However, our current plan is to not move completely out of the bitmap track, but look at output on a case-by-case basis. Using the standard TEX `dvi` files to obtain high resolution output is still the easiest and most cost-efficient means for obtaining camera copy.

Author Submissions

There are many ways for authors who submit TEX files to help a production department handle their files in a quick and economical way. Authors should provide source, macro, and `dvi` files as well as hard copy to the publisher. They should check with the publisher to determine the disk, tape, and other formats required for submission.

Naming of files is important. Authors writing a book a year can title their files `ch1.tex`, etc., but this will force the production department to rename files so they are not lost among all the other `ch1.tex`'s. This applies to macro files as well. Naming your files with the part of the first author's last name should be the standard (e.g., `Spr1.tex`). Do not name files `book.tex`, `preface.tex`, `macro.tex`, etc.

Authors should use the publisher's macros. If they want to modify those macros for their book, they should speak with the production department.

Staff Training

Perhaps the best route for training staff in TEX is to send them to a TEX Users Group training course and then apply that knowledge in-house on numerous projects. In a few short months, a book production department often sees a wide variety of TEX books. This atmosphere provides a rich training ground. Students about to learn TEX should have already acquired strong word processing experience, especially in *search and replace* techniques (macro creation is an added plus). Knowledge of file manipulation is also strongly recommended. TEX creates an enormous number of files. These files must be kept in very specific locations. Simply knowing the difference between subdirectories on a PC or folders on a Mac is without question invaluable.

Springer has also discovered that a varied desktop publishing environment also helps build expertise in TEX. The ability to conduct staff brainstorming sessions in TEX makes a production department an arena for quick learning.

Secondary Usage

The day is very near (if not here already) when the paper product will not be the only form in which a book is published. Standardization of electronic files will become a requirement in the coming years. Authors now have the ability with TEX to e-mail files to colleagues for almost instantaneous interaction over long distances.

SGML is the ultimate standard and may be the goal for production departments when considering the handling of electronic products. This stands TEX in good stead for many years to come. First, it is ASCII-based and easily lends itself to translation to SGML forms. Second, it already is something of a standard in math and physical science departments around the world, giving TEX a great advantage over the proprietary systems of most mathematics typesetters.

Conclusion

Any publisher in the science and technical area, particularly in the mathematical and physical sciences, is going to have to deal with TEX in order to produce cost-competitive books in a high-quality way. Knowledge must be gained in production departments to deal with TEX files, and author–publisher interaction in this area must be supported. Production departments that treat TEX as a black box will not be gaining everything that they can from TEX.