How to Set Up and Maintain a TEX System

R. Allan Reese
Computer Centre
Hull University
Hull HU6 7RX
UK
Phone: +44 482 465296
Janet: r.a.reese@uk.ac.hull.ucc

Abstract

TEX use is increasing, but TEX suffers from a folklore perception of being hard to use and aimed at computer programmers. This is partly because of the packaging and presentation. TEX documentation generally assumes TEX has already been set up — someone else has written a *Local Guide*. Compared with commercial software that is 'load and go', TEX installation seems to expect an unreasonable level of systems knowledge. This is compounded by the lack of a definition of what would constitute a TEX installation; it is far more than just TEX authenticated by passing the TRIP test. The continuing development and improvement of TEX-related tools distributed via networks aggravates the situation. The TEX community must specify the components and the 'current versions' that should be available in any 'standard' installation if TEX is to be used as a *lingua franca* for document transfer. The combination of TUG and CTAN organizations form a reasonable basis for carrying out this function. The paper includes a list of components that this author has assembled over several years; the next stage would be to expand this list and make the list generally available.

The problem

There are several papers and distributed notes on the general theme of 'Getting started with TEX'. Daneliuk (1993) is a list of TEX-related tutorials. Childs et al. (1988) or Martin (1990) describe introductory training in the TEX mark-up language. Doob (1992) is a plain TEX primer, and Warbrick (1992) and Maltby (1992) are primers to LATEX. Rahtz (1992a) is a guide to using (LA)TEX under Unix and discusses related software to put TEX in context. However, all new TEX users are referred to their *Local Guide* for the mechanics of how to use a system that is *assumed to be available*; there is little comparable help for the first person who fetches TEX to a site.

Rahtz (1992b) provides very brief notes for Unix systems, but assumes a lot of systems knowledge and ability on the part of the user if the make files do not work as intended. Luecking (1993) summarizes on email the process and problems of setting up the widely used IBM PC version emTEX, but does not cover what should constitute a 'TEX system'. He writes, "I have assumed that the reader is familiar with the general idea of a TEX system [and that you] understand your setup and how to use DOS."

This paper was presented at the Aston conference as a workshop, which relieved the author of the responsibility of providing definitive answers. Most or all of what it contains may not be new to the reader, but some of it may be rather shocking since it may remind you what you have forgotten.

It deals with a topic that I consider important, because unless we can persuade more people to move from software that is marketed as 'easy to use' (whether it is or not!), (LA)TEX users run the risk of being marginalized — being seen as eccentric, pedantic, weird, 'out of their tree'.

The people. The people gathered at a TUG event will, for the most part, be the frontiersmen (sorry, persons — but I'll stick with the old word and the image) of TEX. They are also likely to over-represent academic users who have the benefit of support over the world-wide Internet. Furthermore, they include people like myself whose jobs are specifically to evaluate, implement and support software for an organization. We can justify the investment of time playing around — or research as we call it on Sundays. Many TUG members would describe themselves as 'computer scientists'. When I first obtained TEX 'for the University of Hull' I was startled to find *several*

copies already on the system, fetched and installed by computer scientists. But their brief does not require them to provide services to others.

The frontiersmen are friendly and hospitable but they may have forgotten the Herculean efforts they endured to get where they are now. Their latest discoveries may look like wild flights of fancy to a newly arrived 'immigrant' used to the pedestrian disciplines of an office-oriented word processor.

British common law is largely based on what is expected of the 'reasonable man' (or woman). What is reasonable to ask of a putative TEX-installer? The following recent exchange on email with an archive maintainer is likely to send the vast majority of current computer users climbing up the wall, or up the tree that we vacated:

> A user asks, "We've been running TeX 3.0. I'd like to get the version 3.14 as on ymir.claremont.edu, but the files are a pain to transfer via ftp ...."
>
> And gets the reply, "If all you're after is TeX, all you should need is TEX.WEB and TEX.VMS-CHANGES, both of which are ASCII files. ... Also there and of interest are VMS_TEX_NOTES.TXT and COMPILE_TEX.COM.
>
> "Get the WEB and VMS-CHANGES files, then TANGLE (probably defined as the foreign command symbol TANGLE:==$TEX_EXE:TANGLE since you're running DECUS-TeX) the files, replying with the appropriate .WEB and .VMS-CHANGES files, also pointing to TEX.PAS as the output and TEX.POO as the Pool file. Once done, run Pascal on your TEX.PAS, then link TEX.OBJ and you should have the executable, TEX.EXE. Alternately, this is done for you in the compile DCL command script. To finish up, COPY/PROT=W:RE TEX.EXE to TEX_EXE and you should be done. The VMS-CHANGES file includes the TEX.CLD file which is distributed with the DECUS files, so the change should be transparent."

Let me stress I'm not criticising the work of archive maintainers. They do a superb job unpaid, and are greatly appreciated by many users including myself. However, TEX may be seen from outside as a club for computer experts, and advice like the above will repel more users than it attracts.

The program. TEX represents an attitude to both typography and computing. It is exciting and innovative. It is iconoclastic, dismissing arbitrary restrictions that are incidental to either the hardware or the system in use. For example, in Chapter 4 of *Computers in Typesetting* (Knuth, 1984), Knuth writes "This is something that authors and readers aren't accustomed to, because printers couldn't do such things with traditional lead types." It is therefore something of a *volte face* by Don Knuth to wish to embalm the TEX program and take it into the afterlife unchanged, or at least to ask that its name shall live for ever and ever. TEX *as used* is constantly evolving and developing, and this represents a problem for anyone who wants to get started.

Knuth's mandate relates only (I think) to the program source of TEX. Any usable 'TEX' system has to be built of many tools and components which are available in a variety of forms and from various sources. While 'mix and match' gives great flexibility and power, it also gives uncertainty and instability. One aspect of 'client-server' computer systems is the split between the front-end that the user sees and the processing engine. Papers at this and previous conferences have described many preprocessors and integrated front-ends that retain TEX as the background processor; there seems almost a fatalism that SGML will be adopted as a document-description language but that its input will be processed into TEX for display.

Most word processors look like Barbie® Dolls. TEX looks like a Meccano® set. The difference is that the doll is a toy with a limited use, and is heavily promoted because buying the doll leads to buying clothes and accessories — like boyfriend Ken. Incidentally, Barbie is anatomically distorted to hyperstimulate human emotional responses — rather like most software hype. Meccano®, for those who don't know it, comes as a kit of metal strips and plates which the user bolts together to build models. The sales pitch is that you are limited only by your imagination and perseverence. You are advised to start with something easy and warned that you may become an enthusiast.

Once again, I'm not criticizing or complaining; I am pointing to the appearance and suggesting that this attracts a certain type of clientele.

The mission. To put the problem into perspective, most programs (for personal computers) arrive as a package of disks and a manual. In the manual is a page or chapter called 'Installation' and usually the procedure involves nothing more than putting 'Disk Number 1' in the drive and typing `install`. Some programs even make changes to your system configuration (I don't like this feature) or indicate possible optimizations. In contrast, TEX looks like, for example,

- 150+ files for PCs from an archive, most of which are compressed, packed sets of files. This

includes multiple versions of most programs, numerous, very technical, `readme` files, and complex environment and configuration variables, or

- an archive file for Unix that expands into 3500+ files in a hierarchy of directories that may not be allowed (by 'Systems') on the target system, or
- the TEX archives themselves, which contain (June 1993) 26000+ files of all vintages and provenances.

While checking this paper I found that a program called `install` has appeared in the emTEX archive; I have not yet tested this feature.

The TEX community, if it is to thrive, must provide guidance for two distinct people (though they might be in one body): the TEX implementor and the user. Both would benefit from having a defined canonical or default 'Current-TeX'. (I defer to Don Knuth on the use of the trademark.) Anyone wanting TEX should be able to get — easily — this definition and obtain the software in a form where it installs as easily as any other product. That is, they don't need to go to several sources, look for FAQs round the world, decypher cryptic in-jokes or take a course in systems programming.

> In current (perhaps just going unfashionable) jargon, fellow TEXies, you have a marketing and image problem.

Moving to a solution

For an individual user, needing one copy of the software, I have no hesitation in recommending they go to a commercial supplier for a TEX package. That's how I got started. The first TEX I used was TurboTEX® and that came as a set of disks and a booklet. Nevertheless, there was considerable room for improvement to match the 'professional', i.e., slick, appearance of other software. Whether the product can be made sufficiently appealing to a mass audience is a commercial decision for the companies.

The corporate user, especially in the impoverished academic setting, has a greater problem. Commercial licences for the packaging make TEX an expense comparable with other word processing software. Advertising and peer-group pressure will incline the naïve (it's compulsory to include *this* word in any TEX document) users towards to most palatable at first sight. Academic software-support staff, unless they are inspired to claw through the undergrowth and fight up the rapids to join the frontiersmen, need far more help in installing the software and writing the mythical *Local Guide* than is currently available.

The user group, and especially the management committee, has a clear rôle to define the 'Current-TeX' standard and to keep an eye on developments so that the standard can be amended or extended by definitive statements, rather than by uncoordinated natural selection. As a start, this would allow archives to be divided in a stable kernel and a seething caldron of experimentation where files can be added *and deleted* in contrast to the current ever accumulating museum of endeavour.

The announcement of the CTAN — the Comprehensive TEX Archive Network (Greenwade, 1993) — at the Aston conference suggests that this is where 'Current-TeX' should be stored. From conference discussions it appears that this should be classified in at least two dimensions:

- as a series of workbenches with appropriate tools — for authors, editors, compositors, style designers, font designers, systems support, etc.
- as a set of levels — entry (absolutely required), advanced, expert ....

As a first stage the standard should be defined in terms of the products and versions that should be available at all installations. This information would be a single document or file. The next step would be to expand this list in terms of actual filenames for particular implementations. This could be done by setting up actual directories with copies of files, but more probably by storing the list of filenames as a document. Many text files would be common to all implementations, while executable binaries would be unique. There is the small but important point of ensuring that files that constitute 'Current-TeX' should not be deleted or changed except deliberately and in concert with changes to the filelist files.

A 'Current-TeX' document would assist site support staff like myself or commercial enterprises who want to sell TEX with 'added value'. At present, the contents do not seem to have changed since *The TEXbook* and *The LATEX User Guide* were written — around 1984. Is the 'new LATEX' (in archives as `latex.tex` dated 25 March 1992) now the standard? It's required for building the NFSS (Mittelbach, 1990) — is *that* now standard?

Defining a kernel

I am not here to prescribe. The following list was compiled from several years of supporting a generally-used system and from reading e-mail discussions. The TEX community as a whole must define what should be included. I'll start the process with

a list of items, an indication of the files involved and a comment on why they should or should not be included. I've used DOS filenames since the extension codes (`.tex`, `.exe`, etc.) conveniently distinguish the type of files.

In some areas, e.g., editors, it is clear that 'Current-TeX' can specify that there must be *an* editor to allow users to manipulate ASCII files. The CTAN can make public domain programs available, but we cannot prescribe which the user will choose.

- TEX
  - `tex.exe`
  - `plain.fmt`
  - `tex.poo`

  TEX as supplied is so overwhelming that it's hard to relate this to the single program described in *The TEXbook*. For most widely-used machines, it would not be hard to maintain a single executable file that needs only copying and running. Barring dramatic problems, this could be updated annually. On the 'level of user' scale, I suggest that the great majority of users will want only the executables, in a form that they can copy and run. Reading and changing the source code is a mark of being 'expert'. BTW: I've never read *TEX, the Program* (Knuth, 1986).

  I was recently asked, "When the archive people talk about TEX 3.1, 3.14, ..., etc., does this refer to changes in the `.EXE` program or the `.FMT` or both?" I confessed that I didn't know. (I think it covers both, but even when changes are made only to `tex.exe`, it is necessary to reprocess `plain.tex` → `plain.fmt`.)

  In several years of using TEX, I've never had to make any changes to the `.POO` file, nor have I heard of anyone doing so. Is this a neglected facility for enhancing one's TEX, or a red-herring?
- LATEX
  - `lplain.fmt`, incorporating
    1. `lplain.tex`
    2. `lfonts.tex`
    3. `latex.tex`
    4. `hyphen.tex` (via `lhyphen.tex`)
  - Standard styles:
    book (`book.sty`, `bk10.sty`),
    report, article, letter, ...
  - Many options:
    page sizes (with samples for parameter names and values), `multicolumn`, `fancyheadings` (should be the default), marginal notes, line numbering, etc.
  - Seminar (let's scrap SLITEX)

It took me a long time to realize the relationship between `lplain.tex` and `latex.tex`. I still don't know where to look up such information. The layered design of LATEX (and flaws in same) were discussed at a meeting of UKTUG in Oxford (February 1993). It would be silly to preempt the LATEX 3 project, but there may be other styles that are widely enough used to be 'canonized'.

As far as I know, LATEX for most systems is still distributed with what is now called the 'old format', and each user has to rebuild the `.fmt` file to the 'new' LATEX dated 25 March 1992 in archives. This is a major potential for problems and inability to exchange documents freely. The *LATEX User Guide* (Lamport, 1986) has a disclaimer on the publisher's page, "Any discrepancy between this description and the behavior of this or any later release of Version 2.09 is an error." Ah, but in the documentation or the program?

I think any supporter of a TEX system has to be able to build a format using `initex` (see below). What should they then do to verify that the new format is conformant? We can assume that they will not take kindly to the suggestion of running the `TRIP` test.

Could the font size options be parameterized? This is more a question for the LATEX 3 project, but the increasing use of arbitrarily scalable fonts (in PostScript) makes TEX's few fixed sizes look old fashioned.

The standard paper size options should cover at least the range offered in `DVIPS`, and should probably incorporate 'best practice'. For example, I'd welcome some advice on good page designs for using A4 paper.

- Writer's tools
  - Simple ASCII editor
  - TEX-aware editor (e.g., emacs, TE — but without the crashes described below)
  - Spell checker
  - TEX-aware syntax checker
  - Word count and other 'editor's tools'

We know TEX is not a word processor, but it is becoming increasingly difficult to buy a simple text editor that is not a word processor. This unfortunately is a double disincentive for many users to consider TEX, firstly as the word processor apparently does what they want and secondly because you have to actively avoid word processor features. You generally have

to take special steps to save WP output as simple ASCII.

TUG could take steps to encourage marketing of commercial and other editors. I know it already does but I get the impression that the typical active TUG member is happy with an environment as complex as `emacs`, which would have most of my university users scrabbling again up that tree. TUG endorsement would add some official weight to complaints such as the two below.

During the Aston conference the editor `redit` became available on the CTAN. This looks attractive but the version available was only partly translated from German. This was actually useful, as running a German editor made it easier (as a non German-speaker) to focus on the structure and appearance.

The amSpell (no connection with $\mathcal{AMS}$) program by Erik Frambach et al. (Frambach, 1992) is a public domain spell checker.

The TeXshell program by Jürgen Schlegelmilch (1993) provides a uniform interface to the various components of a TEX system on a PC. It still requires the local implementor to understand how to integrate each component into the whole.

Two recent problems that have involved me personally in this area have been:

1. the TE editor that can be obtained from archives completely hangs our PCs if told to edit a file greater than 60K. I have not been able to find support or the source of the program
2. I have been promoting the use of Correct Grammar (CG) as a *prompt* to encourage thinking about writing. The program is not TEX-aware and when I contacted the vendors I was told there were no plans to add this feature.

- Tools to build a format file — e.g., for alternative languages

  The ten and a halfth commandment reads, "This is the plain TeX format... And don't modify the file under any circumstances." I put it as ten and a halfth because the better known eleventh commandment takes precedence. The most obvious area where people seem to want to override the contents of `plain` is in font allocation, and in adapting TEX to non-American languages. These are both areas where the average user does *not* want to do the spade-work, but would appreciate being able to be supplied with either simple instructions or, best of all, alternative format files for immediate use. The German TEX supplied with emTEX is perhaps a model for this. There is a TUG working group looking at language problems: perhaps formats for different languages should form part of 'Current-TeX', with English being just one amongst many.

- Fonts — as sets of characters
  - **TEX and LATEX:** Possibly the most FFAQ on fonts is, "Where is `lcircle10`?" TEX fonts are currently in a state of flux caused by the informal nature of TEX evolution. TUG members assume that the DC fonts are now 'standard'; TEX users outside TUG probably would ask, "What's DC?" At Aston it was clear that the only fonts that could reliably be expected in a currently 'standard' installation are those defined by Knuth (1984) and Lamport (1986).
  - **$\mathcal{AMS}$Fonts/:**
    Even those not writing mathematics may require extra symbols. The $\mathcal{AMS}$Fonts/ provide some 200 symbols not in the (LA)TEX sequences. I was made aware of this when a student asked for the three-dot 'therefore' symbol.
  - **NFSS:** As with DC, inside TUG and among users attached to the network and able to download from archives, NFSS is now the standard. Elsewhere?
  - **Viewing tools:** We need to be able to look at and to interrogate fonts, to have a simple way of searching a set of font files for a particular glyph by name. There are `testfont` and a similar macro by Borde (Borde, 1992), but that's it.
  - Tools for converting fonts (e.g., Adobe Type 1 → TEX).

- Fonts — as system files

  `*.TFM, *.PK, *.VF`

  METAFONT is almost certainly needed to avoid trying to store every font at every size. The local supporter maybe only needs to know how to run METAFONT with supplied input, how to handle the results and where to put the output.

  Users are confused and intrigued by the relationship between the printer resolution, design size and character magnification, and the bitmap file. I had to attend a TUG workshop to be clear on this. The documentation on virtual fonts is virtually restricted to the sources of `VPtoVF` and `VFtoVP`. The possibility of `.AFM`

and character mapping fonts for Adobe® fonts also leaves the user bemused.

emTEX's system for packing font bitmaps into library files is commendable, but this leads to yet another extension (`.FLI`) and the need for tools to maintain libraries. A standard for porting fonts would be useful; as binary files they are a pain.

- Printer driver
  There is a lack of information about printer modes, but perhaps this is very technical and esoteric. However, it would be helpful to see some examples of how much (or little) improvement can be achieved by tweaking the parameters. I have installed Mattes' `dvidot` and Rokicki's `dvips` (Rokicki, 1993).
- Previewer
  A fast and versatile screen previewer can make TEX competitive with any WYSIWYG DTP system. Many people (meaning my own prejudice but reinforced by numerous discussions) have indeed commented that immediate design on the screen is less desirable because users who are not professional designers will both see what they expect and accept what they are offered. `dviscr`, `dviscreen` and `xdvi` are all very good.
- PostScript
  PostScript appears to be a world standard for page description as well as a printer driver. At least that's the impression I get, though it's hard to be objective. Some of my colleagues have suggested buying non-PostScript laser printers, and cite HPGL as an equally valid standard. However, in the TEX world, PostScript and Adobe font representations seem a centre of interest. I make heavy use of:
  - PSNFSS
  - PStricks

  (through `seminar.sty`, not as direct calls)
  - Graphics inclusion

  where `bbfig`, the PostScript code that checks `BoundingBox` values is an indispensible tool. I have installed on our system all of the `epsf` macros that come with `dvips` (Rokicki, 1993), the `BoxedEPSF` macros (Siebenmann, 1991) and the `psfig` macros (Darrell, 1987). None is definitive or clearly best.
  - PostScript previewer

  i.e., GhostScript — I use it heavily on DOS and UNIX
- Page manipulating tools
  - dvi manipulation
  - PostScript manipulation

  `dvidvi`, `dviselect`, `dvibook` and their `ps` equivalents
- Formats — on top of `plain` TEX
  - **Eplain:** Expanded Plain TEX (Berry, 1993). Provides tools for cross-referencing and indexing — i.e., LATEX-like facilities without the imposed styles.
  - **Newsletr:** `plain` TEX macros for multi-column working (Goatley, 1991).
  - **TEX by Example** (Borde, 1992).
  - **$\mathcal{AMS}$-TEX:** American Mathematical Society style (Spivak, 1986). This is an *alternative* to `plain`, so $\mathcal{AMS}$-LATEX/ may now be preferred.
  - **texinfo:** Gnu documentation format. Apart from being a format in its own right, this is a stepping stone to a wealth of other excellent software, some of which is TEX-related.
- Formats — on top of LATEX
  - **docstrip:** Another TUG de facto standard, for the distribution of 'documented' LATEX styles. George Greenwade describes it (in email, 1993) as "Literate programming for LATEX."
  - **$\mathcal{AMS}$-LATEX/:** American Mathematical Society options which is an extension standard LATEX. Requires `NFSS`.
- Easy options for verbatim file insertion, insertion of extracts from text files, etc.
- Pre- and post-processing tools
  - **Bibliographic tools:** BIBTEX and Tib (Alexander, 1989)
  - **Indexing:** MakeIndex
  - **Listing:** specific types of source documents (e.g., c2latex)

Whither?

Preparing this review has reminded me how many sources have been tapped to create the TEX system I currently maintain at the University of Hull. Thanks to the many people who have helped me, directly or indirectly.

This paper is an invitation and a challenge to comment to me, to TUG or to the wider TEX community. Even the isolated user with a personal copy of TEX will at some time want to send a document to another TEX user or will want to upgrade their system. Either of these events will cause unnecessary anguish if they are aiming at a wobbly target.

Another necessary aspect of 'Current-TeX' will be consistent and accessible documentation. One way forward will be to gather *Local Guides* from many sites, to compare what is made available and how it is accessed. It should be a fair assumption that *Local Guides* are themselves written in (LA)TEX. If you have written a document, please deposit a copy with the CTAN.

What's an archive?

The provisional recommendations here contain several items that are commonly distributed through the TEX world-wide archive network. These are then retrieved *in their instantaneous incarnation* by file transfer or email. If the 'Current-TeX' concept is taken up by TUG, this is not a satisfactory way of providing a consistent system for standard sites. From discussions at Aston, it seems that those who maintain archives are sympathetic to the views expressed in this paper. Various technical methods for keeping archives internally consistent were suggested and will be further discussed.

Bibliography

Alexander, James C. "**Tib** A TEX Bibliographic Preprocessor. Version 2.2." Public domain software available from TEX archives. 1989.

American Mathematical Society. "$\mathcal{AMS}$-LATEX/ Version 1.0 User's Guide." 1990.

Berry, Karl. "Expanded Plain TEX—Version 2.3." Public domain software available from TEX archives. March 1993.

Borde, Arvind. *TEX by Example: A Beginner's Guide.* Academic Press, 1992.

Childs, Bart, et al. "Syllabi for TEX and METAFONT Courses." *TEXniques* 7, Pages 117–128, 1988.

Daneliuk, Tim. "List of TEX-Related Tutorials as of 05.18.93." email available from `info-tex@shsu.edu`, 1993.

Darrell, Trevor. "`psfig` Version 1.8." Public domain software available from TEX archives, 1987.

Doob, Michael. *Gentle Introduction to TEX.* Public domain document available from TEX archives, or printed form from the American Mathematical Society, 1992.

Frambach, Erik, et al. "amSpell—Version 2.03." Public domain software available from TEX archives, 1992.

Greenwade, G. "The Comprehensive TEX Archive Network." *TUGboat* **14**(3), in press, 1993.

Goatley, Hunter. "`NEWSLETR—plain` TEX Macros for Newsletters." Public domain software available from TEX archives, 1991.

Knuth, Donald E. *The TEXbook.* Volume A in *Computers and Typesetting.* Reading, Mass.: Addison-Wesley, 1984.

Knuth, Donald E. *TEX, the Program.* Volume B in *Computers and Typesetting.* Reading, Mass.: Addison-Wesley, 1986.

Lamport, L. *LATEX: A Document Preparation System.* Reading, Mass.: Addison-Wesley, 1986.

Luecking, Don. "Setting up emTEX." Information file available from TEX archives, 1993.

Maltby, Gavin. *An Introduction to TEX and Friends.* Public domain document available from TEX archives, 1992.

Martin, C. R. "TEX for TEXnical Typists." *TUGboat* 11 No 3, Pages 425–428, 1990.

Mattes, Eberhardt. "emTEX." Public domain document available from TEX archives, 1993.

Mittelbach, Frank, and Rainer Schöpf. "The New Font Family Selection." *TUGboat* 11 No 1, 1990.

Rahtz, Sebastian. "A LATEX Survival Guide." Notes distributed with Sun Sparc implementation of TEX, 1992a.

Rahtz, Sebastian. "System Guide for Setting Up TEX on a Sparc." Notes distributed with Sun Sparc implementation of TEX, 1992b.

Rokicki, Tomas. "DVIPS 5.519" Public domain software available from TEX archives, 1993.

Schlegelmilch, Jürgen. "TEXShell—Version 2.5.2." Public domain software available from TEX archives, May 1993.

Siebenmann, Laurent, "`BoxedEPSF.TEX`." Public domain software available from TEX archives, May 1991.

Spivak, Michael. *The Joy of TEX.* Addison-Wesley, 1986.

Warbrick, Jon. "Essential LATEX." Public domain document available from TEX archives, 1992.

Van Zandt, Timothy. "PSTricks—PostScript Macros for Generic TEX." Public domain software available from TEX archives, 1992.

Van Zandt, Timothy. "`seminar.sty`—A LATEX Style for Slides and Notes." Public domain software available from TEX archives, 1992.