

---

**Book review: *Writing With T<sub>E</sub>X and T<sub>E</sub>X & L<sup>A</sup>T<sub>E</sub>X: Drawing & Literate Programming*, by Eitan Gurari**

Michael D. Sofka

Eitan M. Gurari, *Writing with T<sub>E</sub>X*. McGraw-Hill, New York, 1994, ISBN 0-07-025207-6.

Eitan M. Gurari, *T<sub>E</sub>X & L<sup>A</sup>T<sub>E</sub>X: Drawing & Literate Programming*. McGraw-Hill, New York, 1994, ISBN 0-07-911616-7.

In this pair of books Gurari has written primers for using T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X for writing, programming and drawing figures. The books are short and to the point, with most commands being introduced and explained in one or two paragraphs. If you are new to T<sub>E</sub>X, or do not like to experiment, these books may not be for you. If, on the other hand, you are looking for a short and comprehensive review of T<sub>E</sub>X, or if you want to draw figures and write programs in T<sub>E</sub>X, Gurari is a good choice.

*Writing With T<sub>E</sub>X* is a T<sub>E</sub>X reference manual. It starts with a three page *Getting Started*<sup>1</sup> chapter which explains how to run T<sub>E</sub>X. Most of the remaining book is divided into two parts. Chapters 2–10 cover *T<sub>E</sub>X as a Formatting Language*. This includes simple text, *Fonts, Layout of Pages* and *Mathematical Formulas* (each in five pages!), inserts, boxes, and basic programming constructs such as *Groups* and *scanning*. The concepts and commands introduced, however, are only the essentials needed for formatting.

The second part consists of chapters 11–16 and is called *T<sub>E</sub>X as a Programming Language*. It covers *Macros, Data Types, Selectors* and *Auxiliary Files* (file I/O), each of which is a basic programming unit. These chapters are a little longer, averaging ten pages each. The programming is rounded out with two chapters covering macros and character codes in more detail. This includes `\let`, `\edef`, `\csname... \endcsname`, token expansion, active characters, changing character codes and other subtleties of writing T<sub>E</sub>X macros.

Chapter 17 is titled *Environments For Writing*. Macros are described for basic document structures such as `\Chapter` and `\Section`, as well as environment blocks, lists, cross-references, table of contents, figures and bibliographies. L<sup>A</sup>T<sub>E</sub>X<sup>2</sup> is introduced at the end of this chapter as an example style library.

---

<sup>1</sup> When chapter titles appear as part of a description they will be italicized.

<sup>2</sup> Version 2.09

The book ends with eight appendices. The first six are divided by tasks and cover T<sub>E</sub>X commands in more detail. Tables and output routines are addressed here, as well as a more about symbols, boxes and penalties. Appendix G is a complete catalog of T<sub>E</sub>X commands, each with a short explanation and example. The final appendix is a short bibliography of T<sub>E</sub>X books, newsletters and electronic resources.

I have held the opinion that as a general rule a programming language should be describable in about 100 pages. T<sub>E</sub>X is complex enough that a short reference is difficult. In *Writing With T<sub>E</sub>X*, however, Gurari has made a decent 230-page attempt. This economy of language, however, comes at the price of ease of learning and depth of knowledge. If you have already learned the basics of T<sub>E</sub>X and wish to know more, there are other books which go into the details of macros, line and page breaking and output routines. If you are new to T<sub>E</sub>X, and not skilled with programming, Gurari offers little help. On the other hand, *Writing With T<sub>E</sub>X* is a decent T<sub>E</sub>X reference. Gurari's explanations are clear, concise and independent, which is what a reference manual should be.

*T<sub>E</sub>X & L<sup>A</sup>T<sub>E</sub>X: Drawing & Literate Programming* is a reference manual for two macro packages written by Gurari: DraT<sub>E</sub>X for drawing pictures, and ProT<sub>E</sub>X for literate programming. The book title is misleading since it is really about drawing and literate programming (and specifically about DraT<sub>E</sub>X and ProT<sub>E</sub>X). Very little of the book deals generically with T<sub>E</sub>X or L<sup>A</sup>T<sub>E</sub>X.

The book begins with a short *Getting Started* chapter which explains how to run T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X. Chapters 2–5 cover the basics of T<sub>E</sub>X (including *Mathematical Formulas* and *L<sup>A</sup>T<sub>E</sub>X*) in a whirlwind 20 pages. Given how peripheral this is to the actual macro packages, I wonder why these chapters are included — it is unlikely somebody will purchase *T<sub>E</sub>X & L<sup>A</sup>T<sub>E</sub>X* who is not already familiar with T<sub>E</sub>X. Still, Gurari is good at providing short and accurate explanations, and I dare say the five page L<sup>A</sup>T<sub>E</sub>X chapter is enough to get started.

Chapters 6–11 cover DraT<sub>E</sub>X primitives. These include macros for line and curve drawing, painting and clipping, coordinate systems (including three-dimensional viewpoints), repetition and data paths, creating objects with tables of data, and arithmetic operations. DraT<sub>E</sub>X is indeed an impressive package which does all of its drawing in T<sub>E</sub>X. This does, however, impose some limits on what can be drawn. For example, rotated text is not possible without special fonts or `\special` support, and the clipping and three dimensional commands are limited.

Chapters 12–20 cover *High-Level Drawing Facilities*. These are drawing templates provided by `AlDraTeX`, a macro package based on `DraTeX`. The templates include macros for pie, XY and bar charts. There are three chapters introducing diagramming, a chapter on tree diagrams and another on labeled graphs. Gurari has authored a textbook on computational theory, and the graphs and trees provided by `AlDraTeX` seem sufficient for such a book.

Chapters 21 and 22 cover *Literate Programming* using `ProTeX`.<sup>3</sup> Literate programming is a programming style invented by Donald Knuth, and used for writing `TeX`, `METAFont` and the suite of programs used for font management. Literate programs consist of a mixture of code and code documentation. A set of filters convert the input file into either the source code for the program, or a typeset document describing the operation of the program. Gurari's `ProTeX` is a macro package which uses `TeX` as that filter. `ProTeX`'s output is a typeset document and auxiliary files consisting of the code described in the document. The code may be Pascal, C or any other language including `TeX`.

The book ends with five appendices. The first two cover the *Implementation of ProTeX*, and the use of *PostScript Figures Within TeX*. `ProTeX` implementation details are provided so that users can adapt `ProTeX` to their own needs. `PostScript` is introduced as a standalone page description language, and as a way to supplement the abilities of `DraTeX`. In this mode it is similar to Timothy van Zandt's `PSTricks`.

Appendix C is a *Catalog of Commands* covering `TeX`, `LATeX`, `DraTeX`, `AlDraTeX` and `ProTeX`. Each command is presented with an example of its output. It would be helpful if the commands were cross-referenced with their explanation in the book.

The last two appendices are a bibliography and information on acquiring the macro packages. The bibliography includes references on drawing and literate programming. The macros are available on a disk which accompanies the book, but updates and examples are online.

Regardless of the systems on which you work, there are a variety of affordable (frequently free) drawing programs available. For complex figures these are usually a better choice. For simple figures, or if you like the idea of providing a single,

portable source document containing both text and figures, `DraTeX` and `AlDraTeX` are a good choice.<sup>4</sup>

`DraTeX` and `ProTeX` are available on CTAN and at `ftp.cis.ohio-state.edu` in the directory `/pub/tex/osu/gurari/` (they have been updated since the book and disk were published). This includes an `Examples.tex` file, which is a `ProTeX` file of examples from the book. There is no better way to sample the abilities of `ProTeX` and `DraTeX` than to compose this file. The result is a typeset description of the examples, and 68 example figures and exercises.

On the whole, both books are well written and cover a lot of material in few pages. There are, however, no answers provided for the exercises so expect to spend time experimenting. I recommend these books only for experienced programmers who need a concise `TeX` reference manual, or who would like to use a `TeX` based drawing or literate programming package.

◇ Michael D. Sofka  
 Computing Information Services  
 Rensselaer Polytechnic Institute  
 Troy, New York 12180-3590  
[sofkam@rpi.edu](mailto:sofkam@rpi.edu)

<sup>3</sup> This is the third package I am aware of called `ProTeX`. The other two are *Professional* typesetting systems based on `TeX`.

<sup>4</sup> You will, however, require a version of `TeX` compiled with a large main memory array, and a fairly fast computer. I ran the samples on an RS/6000-250 with `Web2c`, and a 100Mhz Pentium with `MikTeX`. On both systems performance was acceptable.