
What is TeX?

Douglas Waud

This journal, *TUGboat*, first appeared in 1980 and has been going strong since. It provides a continuous stream of articles which inform the reader on the latest developments in the field. And there is the problem—the journal has been in production so long that readers are often assumed to know the context. This automatically provides a barrier to an outsider who might stumble onto the journal and ask “What’s going on here?”

The editors have therefore decided periodically to insert a primer, a brief summary, to let such a newcomer know what we are so excited about.

1 So what is the object of our affections?

The simple answer is ‘TeX’. This is formally called a typesetting program, computer software for producing nicely printed output. TeX came about when Professor Donald E. Knuth, at Stanford University, was planning to publish a series of books on computing. He discovered that the classical process of typesetting books was no longer viable and its various replacements were rapidly declining in quality—and so he decided to take a year off to rectify that omission. He got hooked and the year turned into ten. The centerpiece of his final solution was the program TeX. In a nutshell, TeX allows ordinary people to produce beautiful output of top quality.

Now, one unfamiliar with TeX may think “But I can do that with my word processor!” However, that is a misconception. First, the output will generally be a notch below that of TeX. The situation is analogous to good cooking; you may not know what you are missing until you get a taste of a top quality product. The word processors are getting better these days (for example, they are starting to incorporate many of TeX’s tricks) but there is still catching up to do.

There is still another significant distinction. A word processor is described as “WYSIWYG”. This is an acronym for “What you see is what you get”, with the implication that, as you type, the end result will appear precisely on the screen in front of you. The TeX community interprets WYSIWYG tongue-in-cheek as “What you see is *all* you get!” The point is that, once you get spoiled by TeX, you will not be satisfied with anything less than perfection.

The WYSIWYG issue underlies an even more profound theme in TeX. There is a clear division of labor. The job of deciding what you want to say is

separated as much as possible from that of deciding what final form it should take. This has several advantages. For example, as I am typing this, I pay no attention to how it will finally come out on the page. That will be determined by TeX and ancillary programs written by people who are far more competent than I to produce an esthetically optimal final result. Now, if I have strong views, special training, or an area off the beaten track where what I want is just not available (rare these days) I can tweak to my heart’s content. But I don’t have to. If I simply want to “get the job done”, I can do just that without distractions as to how everything will or should be printed. The only formatting I do is tell the program things like “This is what the title will be”. In the present case, all I had to do was type ‘\title{What is \TeX?}’ which, incidentally, shows you how easy “talking TeX” can be; even if you know nothing whatsoever about TeX, I think you can read that segment of code. You have one foot stuck in the flypaper already!

Now we come to the jewel in the TeX crown—its ability to typeset mathematics properly. In this area there is no competition. The American Mathematical Society, in fact, was one of TeX’s principal sponsors; the vast majority of mathematicians, physicists, and other scientists write their papers using TeX.

When TeX was first developed computers were rather primitive compared to today’s. Knuth recognized this and built in modularity. Thus, when TeX processes a file, it does not know what sort of printer or display will be used to view the result. Knuth therefore had TeX put its output in a general form which could then be used as input by “drivers” designed to talk to whatever new printer or monitor came along. This intermediate form is called a “device independent file” (with the file extension .dvi). In turn, there are file viewers, like xdvi in GNU/Linux, for displaying such files on a computer screen.

2 Now meet the family

So far, I have been using the term TeX in its original context—the program Knuth wrote to carry out the magic. However, nowadays, the term “TeX” can also imply a whole family of related programs. TeX is just the grand-daddy of a family of related tools. We can list the main cast now.

TeX First, TeX itself has gone through several updates. TeX78 came first, TeX82 next, and the current and final version is TeX90. (You don’t

really have to know these details since you will only use the final version now.)

Metafont Next, a critical part of typesetting is the fonts used. Therefore Knuth also pursued the design of typefaces for $\text{T}_{\text{E}}\text{X}$. The result was a companion program `METAFONT` which can be used to produce tailored characters. (Again, this is more for information; you will not be jumping into font design your first day!)

L^AT_EX Now we come to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. This is a component designed to shield the author from the details of $\text{T}_{\text{E}}\text{X}$. Leslie Lamport, the author of $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, recognized that, although $\text{T}_{\text{E}}\text{X}$ is very accessible for the sort of person who loves to wallow in things computer, it can be very formidable to, shall we say, more normal folks. Thus he put together a simpler system designed to protect the author still further from the details of the underlying machinery. For example, he allows the author simply to put something that is to be written exactly as it is typed between paired `begin` and `end verbatim` commands and allows the author to completely avoid having to make any decision as to what type face to use, what size, what style, and the like.

History: The original version, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2.09, has been replaced by $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2\text{e}$.

BibT_EX `BIBTEX` is an ancillary program to help organize bibliographic references.

MakeIndex `MakeIndex` is another ancillary program, this time for facilitating of creation of an index.

dvips Earlier we mentioned `.dvi` files as a generic form of output. With time, the need for a general file format was recognized more widely and the company Adobe Systems Inc. created the PostScript language. The $\text{T}_{\text{E}}\text{X}$ community responded with programs like `dvips` to convert `.dvi` files to `.ps` files.

pdfT_EX and pdfL^AT_EX The next step is where we are currently. PostScript has been augmented with the “Portable Document Format” (PDF), also a creation of the folks at Adobe. If you have used the Acrobat reader to read a file, you have been looking at a `.pdf` file. The $\text{T}_{\text{E}}\text{X}$ community has kept up with this development with the program `pdfTEX` and its $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ variant `pdfLATEX`. These allow one to use tricks available in the `.pdf` format, in particular, to create “hyperreferences”, links to stuff on the web.

3 What next?

At this point I encourage you to get your feet wet. Specifically, try to create some documents in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (easier than pure $\text{T}_{\text{E}}\text{X}$ but not out on the “bleeding edge” enough to be overwhelming). First I would recommend getting a manual. There are many out there but I would suggest you start with Lamport’s original guide [2]. Next you need to have a $\text{T}_{\text{E}}\text{X}$ system on your computer. If you use Linux, life is easy; your distribution will probably already have it installed. If you use Windows or a Macintosh I suggest you start with Flynn’s excellent introduction [1]. If you do not have access to back copies of *TUGboat* you can get it online from:

<http://www.tug.org/tex-archive/info/beginlatex/html>

where, in particular, the chapter “Installing $\text{T}_{\text{E}}\text{X}$ ” will be a succinct guide to getting a system installed.

One final parting word of advice: start with something simple, perhaps just try an example from Lamport’s book, and then add frills one at a time, *i.e.* crawl before walk!.

4 Online column

At the same time they are launching this column, TUG is planning an online journal which will include a column “`\begin{here}`” which will be directed toward the person first trying to come to grips with $\text{T}_{\text{E}}\text{X}$. This column will attempt to identify and clarify standard stumbling blocks. This online journal will be available at

<http://www.tug.org/pracjourn>

References

- [1] Peter Flynn. Formatting information. *TUGboat*, 23(2):115–237, 2002.
- [2] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, MA, 2nd edition, 1994.

◇ Douglas Waud
 Department of Pharmacology
 University of Massachusetts
 Medical School (retired)
 17 Lantern Lane, Shrewsbury, MA,
 USA
douglas.waud@umassmed.edu
<http://users.umassmed.edu/douglas.waud/>