

Travels in \TeX Land: A Macro, Three Software Packages, and the Trouble with \TeX

David Walden

Abstract

In this column in each issue I muse on my wanderings around the \TeX world. In this issue, I deal with three *unrelated* topics: I describe how a small macro works that I decided to try to understand, I briefly describe my experiments with three \TeX -related software packages, and I give my perspective on why lots of people find \TeX difficult.

1 Understanding a small macro

When I need some feature of \LaTeX or \TeX that I don't already know, I usually just find an example of what I want to do and copy it without understanding it. If I need to change it, I do it hit and miss, still mostly without understanding. However, occasionally I decide to try to understand an example I have found. In these cases I like to write up what I've learned, to help me be sure I've really got it, and, who knows, it may help out someone else, too.¹

For years I heard the words “kern” or “kerning” without knowing what they meant. More recently I understood it meant something having to do with the tightness of spacing of letters in a word. Recently I saw a simple macro — a definition for typesetting the \LaTeX logo that involves kerning — and decided that it was time to actually understand kerning. This example had such an extreme example of kerning that I figured the example would make clear how kerning works, and perhaps I'd also learn some other things. Here's the example:

¹This is a pretty fragmented way to learn stuff, but it is mostly the way I do it.

```

\def\LaTeX{%
  L\kern-.36em
  {\setbox0=\hbox{T}%
   \vbox to \ht0{\hbox{\the\scriptfont0 A}\vss}}%
  \kern-.15em
  \TeX
}

```

Of course, the above example is the simplest kind of \TeX macro definition. The `\def` command says that the control sequence `\LaTeX` is to be replaced with the text between the open brace at the end of the `\def` line and the matching close brace six lines later to produce the iconic version of `LaTeX`, i.e., \LaTeX .

The first letter of the replacement text is the letter `L`, which is obviously a good start. The next replacement text is `\kern-.36em`. To see what the `\kern` command did, I defined the macro

```

\def\La{L\kern-1em%
  a
}

```

Calling this macro resulted in

aL

In other words, `\kern` followed by the value `-1em` tightens the spacing between the letter `L` and the letter `a` so much that the `a` is moved all the way to the left side of the `L`. The `em` units are traditionally about the width of a capital letter `M` in the current font.² Negative values of `\kern` tighten the spacing between letters and positive values widen the spacing.³ No doubt someone did a lot of trial and error with values for `\kern` to choose the value that moves the letter `A` to have just the right amount of overlap with the letter `L`.

Looking at how \LaTeX prints, it is clear that the next part of the `\LaTeX` macro, i.e.,

²Paul Abrahams et al., *TeX for the Impatient*, Addison Wesley, 1990, page 60 (also available free on the web from <http://www.tug.org/ftp/tex/impatient/book.pdf>). However, the `em` units, in fact, are specified as part of every font and even can be changed by the user with the `\fontdimen` command.

³This is how `\kern` works in \TeX 's horizontal mode; in vertical mode it does something different.

```
{\setbox0=\hbox{T}%
  \vbox to \ht0{\hbox{\the\scriptfont0 A}\vss}}%
```

in some way moves the letter A up and makes it smaller. Here's how that works.

- A box in T_EX is a two-dimensional shape having a height above the Baseline, a depth below the Baseline, and a width from a Reference point.⁴ T_EX makes pages by “gluing” boxes containing individual characters together into bigger boxes that in turn are glued together into bigger boxes until a page is filled. The bigger boxes can be hboxes (for horizontal sequences of characters) or vboxes for vertical stacks of hboxes.⁵
- T_EX has 256 box registers in which boxes can be saved.⁶ Thus, the construction `\setbox0=\hbox{T}` sets the value of box0 to be a horizontal box with the single letter T in it.
- `ht0` has the value of the height of box 0, i.e., the height of a letter T.⁷
- The construction `\vbox to \ht0{...}`⁸ creates a vbox whose height is the value the height of box 0 and whose content is an hbox containing the stuff within the braces.⁹
- `\scriptfont0` is an identifier for script type font 0. Juxtaposing `\the` in front of `\scriptfont0` selects the specified font, i.e., it is as if the control sequence for the specified font had been written. So, `{\the\scriptfont0 A}` says that the letter A should be in the format of `scriptfont0`.¹⁰
- The `\vss` command that is also within the vbox adds enough vertical space to the vertical box to fill it up exactly,¹¹ since T_EX doesn't like boxes to have

⁴According to chapter 11 of Donald Knuth's *The T_EXbook*.

⁵To be more precise, hboxes and vboxes can be nested arbitrarily.

⁶Ibid, page 120.

⁷Ibid, page 120.

⁸Ibid, page 77.

⁹Ibid, page 80.

¹⁰Ibid, page 153.

¹¹Ibid, page 72, and *T_EX for the Impatient*, page 158.

too little or too much “stuff” in them.¹²

- Thus, that letter A in that font, which apparently is a little font, is placed by the `\vbox` at the height of a letter T as we see is in the `\ET` part of `\LATEX`.

Returning to the whole macro

```
\def\LaTeX{%
  L\kern-.36em
  {\setbox0=\hbox{T}%
   \vbox to \ht0{\hbox{\the\scriptfont0 A}\vss}}%
  \kern-.15em
  \TeX
}
```

we see it ends by decreasing the tightness of the spacing using another `\kern` command, and then the `\TeX` macro¹³ is called producing `\TeX` to complete `\LATEX`.

2 Three software packages

I recently tried three new (to me) `\TeX`-related software packages, installing each on my Windows 98 system.

Pro`\TeX`t

I hadn't upgraded my MiK`\TeX`-WinEdt-Ghostscript-Adobe-Acrobat configuration in at least seven years. So it seemed about time in mid-May to try to get things up-to-date. Until I read the last issue of *TPJ*, I had always wondered what to do with the CDs and DVDs that come yearly from TUG. However, in the last issue, there was an “Ask Nelly” answer from Karl Berry explaining, among other things, what the pro`\TeX`t CD is about. So, I went to Karl's answer again and there found a link to the pro`\TeX`t home page on the TUG web site. That told me to start the pro`\TeX`t CD

¹²An experiment of leaving out `\vss` showed no visible change in the resulting printout; but Karl Berry told me that, in this case, its purpose is to avoid an underfull or overfull box message in case the height of A is different than the height of T.

¹³*The `\TeX`book*, page 66.

and follow the PDF-based instructions on the CD. I got the instructions, printed them out, and read them (as they recommend, although it is not my usual practice to read directions).

The next morning, I again loaded the pro \TeX t CD and began to follow the directions. The directions consist of blocks of explanatory text and instructions and places to click to move to the next step in the installation process. First the directions said to delete my old MiK \TeX , WinEdt, Ghostscript, and Ghostview installations. I did this although it was pretty scary—I would have nothing to go back to if the installation of the new stuff failed.

Next, I installed MiK \TeX to my hard drive but I ignored its default “Large” option and instead selected the “All” option, which took 2.5 hours to load and install. I was pleased that it allowed me to put the root $\text{\textbackslash texmf}$ hierarchy on my C drive where I keep applications and to put my $\text{\textbackslash localtexmf}$ hierarchy on my E drive where I keep application files. The instructions also tell you how to change some MiK \TeX configuration files to handle 8 1/2 by 11 paper, and that was no problem. They didn’t mention that I needed to set up my printer offsets for dvips which I figured out by bashing around in the dvips configuration file.¹⁴

The installation process allows you to install \TeX nicCenter or WinEdt and three of its plugins. Since I already used WinEdt, I chose it and clicked to install it. These days WinEdt is supposed to automatically find and link to MiK \TeX , Ghostscript, etc. Its installation also was supposed to preserve my WinEdt registration code. It did find the other applications, but it did not find my old registration code (perhaps my deletion of my old WinEdt files had been a little too brute force); fortunately, I still had a copy of the email message with my WinEdt registration, but I had to remember how to give it to WinEdt since this is not assumed by the pro \TeX t instructions. After a day of use, I also had to bash around in WinEdt and its configuration files for an hour or more trying to figure out how to tell WinEdt to use the latest Adobe Reader on my machine when I click the Adobe Reader icon rather than using my Acrobat 3.0 version of Exchange to display PDF files.¹⁵

¹⁴I found the following parameter line in the dvips configuration file by searching for the word “offset”: 0 0pt,50pt. The capital letter 0 apparently stands for “offset,” and I printed out lots of trial sheets of paper as I tried many values in the rest of the command line before settling on zero and fifty as good enough.

¹⁵There are no general lessons to be learned by me describing the completely ad hoc path I stumbled along until I managed to point WinEdt at the latest Adobe Reader. It is better to hope that WinEdt’s automatic configuration capability does the right thing.

All in all, the installation went relatively smoothly, did not take too many hours (compared with various worse case scenarios I imagined), and I again have a working, much more up-to-date environment for doing my $\text{T}_{\text{E}}\text{X}$ work. I do think the $\text{proT}_{\text{E}}\text{Xt}$ instructions could address a few more of the possible eventualities (but maybe it is better to keep them simple). If I was a brand new $\text{T}_{\text{E}}\text{X}$ or WinEdt user, it would have probably taken me a lot longer to figure out how to make the several detailed changes I had to make that were not covered in the $\text{proT}_{\text{E}}\text{Xt}$ instructions.

Win $\text{T}_{\text{E}}\text{X}$

When someone suggested doing a 30-day trial of Win $\text{T}_{\text{E}}\text{X}$ (<http://www.tex-tools.de/>), I jumped at the chance. Although I didn't intend a systematic or deep evaluation, I was interested in it, based on the developer's web site description of it as a " $\text{T}_{\text{E}}\text{X}$ / $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ editor with MS Office look and feel."

The download and installation worked just fine, and I used Win $\text{T}_{\text{E}}\text{X}$ on a few little things for a week or so. It all worked quite smoothly with lots of nice $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ structure and features visible to minimize the amount of $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ keyboarding needed (and keyboarding mistakes). Win $\text{T}_{\text{E}}\text{X}$ did seem a little slow to launch compared with my usual text editor. I didn't dig deeply enough to understand how powerful the Win $\text{T}_{\text{E}}\text{X}$ editor is (e.g., whether it has regular expression searches). All in all and given the minimal trial I did with it, it looks like a plausible choice for people using $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

I do have one gripe. Win $\text{T}_{\text{E}}\text{X}$ captures the meaning of one (or more?) file extensions, .tex in particular. Now this is not something I can specifically hold against Win $\text{T}_{\text{E}}\text{X}$ —most Windows applications seem to do this. And while I was testing Win $\text{T}_{\text{E}}\text{X}$, it was not so bad that when I clicked on a .tex file, Win $\text{T}_{\text{E}}\text{X}$ launched. However, after I uninstalled Win $\text{T}_{\text{E}}\text{X}$, clicking on a .tex file gave an error message saying the application couldn't be found. It is not too hard to reassign another program to the .tex extension if you know how, but for me it was a struggle to figure out how again. I wish Windows application trial installations would not capture any extension that might be used by another application and would leave doing this until an explicit user command has been given.

Word2T_EX

I have a project where I need to convert twelve lengthy papers that their authors wrote in MS Word into L^AT_EX for inclusion in a collection of papers for which I am using L^AT_EX to typeset the volume. Thus, I looked around (again) for a program to automatically convert Word to L^AT_EX and found Word2T_EX.¹⁶ The Word2T_EX web site (<http://www.chikrii.com>) has some decent testimonials for Word2T_EX; and I also asked *TPJ* editor Lance Carnes about it since his company (www.pctex.com) is a reseller of Word2T_EX, and he told me he has heard good things from his customers. So, I downloaded and installed the 30-day trial of Word2T_EX, which severely limits the number of equations, figures, and tables that will be converted from one Word file. The installation went smoothly.

All one does to convert a document from Word to L^AT_EX is open the .doc file with Word and then save it specifying the .tex file type, and a L^AT_EX file is saved. My test case was a 25-page document with no math, three figures, and no tables, so the artificial limits of the trial version of the program did not get in the way much. The conversion was quite sensible, although the generated L^AT_EX was a little wordy. The resulting file was well-formed L^AT_EX that compiled immediately from L^AT_EX to a .dvi file that I could view with my previewer. I did have to make some changes to the resulting L^AT_EX to change what I wanted the output to look like that was different than the Word file looked.

I don't know what would have happened if I had pressed Word2T_EX to do harder tasks. It did seem sufficient for my task, and I gave them my credit card number and ordered a copy that would not have the artificial limitations the trial version had.

3 The trouble with T_EX

For the last issue of *TPJ*, Arthur Ogawa wrote a long opinion piece on some of the problems with the T_EX interfaces ("In my opinion: T_EX's Interface Challenges, *The PracT_EX Journal*, issue 2, 2005). Arthur presented a sophisticated analysis of what could be done to make T_EX easier to learn and use. I have a more primitive take on why new users of T_EX, L^AT_EX, and the rest find T_EX hard to learn.

¹⁶For more Word-to-T_EX conversion options, see <http://www.tug.org/utilities/texconv/>.

First, T_EX has become very fragmented with lots of distributions, styles, packages, platforms, etc.¹⁷ Of course, such capabilities are also part of T_EX's power.¹⁸ While there are on-going efforts to do some consolidation (such as creating more standard ways of using CTAN), I see it as inevitable that there will be to-some-extent competing versions of T_EX available and a vast array of sometimes conflicting add-ons.

Second, T_EX looks "different" to typical word processor users because WYSIWYG and Word have won the word processing game (at least for now).¹⁹ T_EX didn't win the typesetting game either,²⁰ probably because it doesn't work in the way that professional typesetters are used to — doesn't let them fuss with the sorts of things they like to fuss with. Despite many efforts to make T_EX more graphical to use (e.g., Scientific Word, BaKoMa T_EX Word, WinTeX, LyX, the WinEdt "table designer," etc.), I think much T_EX use will remain command driven with distinct edit and compile steps.

Third, most people never want to learn more than the minimum about anything they have to use or do. We learn to use a couple of buttons on our microwaves but no more. We go to one supermarket and not others because we know where stuff is in the first supermarket (and we are annoyed when the supermarket rearranges the aisles). We don't know how to use most of the dials and controls on the dashboards of our cars. Etc.

WYSIWYG and Word (i.e., the point-and-click GUI interface) have won in the word processing world partly because one can use it like a typewriter without knowing anything other than how to save a file with a new name. Most people I

¹⁷Such fragmentation typically happens in every field after a successful innovation. New companies are started with different or better versions of the initial innovation, many after-market options are available, etc.

¹⁸I might argue that it is T_EX's astonishing interoperability of various components, releases, formats, etc., that allows diverse distributions (the fragmentation I am referring to) from different vendors and providers that T_EX users can move between rather than there just being different products that may not interoperate so well as is the case with normal product fragmentation. In other words, all that good stuff provides part of the complication we see ourselves faced with.

¹⁹This also typically happens after a successful innovation and its following fragmentation — next comes consolidation around one or two offerings, and everyone else eventually goes out of business or hangs on with minor market niches.

²⁰Which, apparently, has been substantially lost to systems like Quark.

know, whether they are PhDs in science or home users who know next to nothing about computers, use Word like a typewriter. They tab and space to center a line, and they tab to do indents including nested indents; they have no knowledge of the ruler-bar for setting margins and first line indents. They do footnotes without knowing about automatic numbering or even how to have them automatically appear at the bottom of the same page as the footnote reference. They find and edit by using the arrow, delete, and backspace keys; they can use Cut and Paste on the menu but don't know anything as simple as `control-X` to cut and `control-V` to paste. And so on. People use Word because it is the standard, and they don't have to learn anything and prefer their primitive known ways of doing things to learning anything new.

To reiterate, people generally don't want to learn anything new.²¹ Of course, with $\text{T}_{\text{E}}\text{X}$ you can't know so little.

The key problems, as I see them, to $\text{T}_{\text{E}}\text{X}$ having any chance of gaining new popularity are that $\text{T}_{\text{E}}\text{X}$ requires massive consolidation, simplification, and a return to being perceived as the state of the art for word processing and typesetting. Unfortunately, this would require a development effort that is too big to have much chance of getting started and less chance of being completed successfully.

Yet, there are lots of users who chose to use $\text{T}_{\text{E}}\text{X}$ and stick with it for some reason: (a) we need its power, (b) a thesis supervisor insisted, (c) we can't abide Microsoft or using Word, (d) using or supporting $\text{T}_{\text{E}}\text{X}$ somehow provides us a living, and, (e) a few may just love intricacy or $\text{T}_{\text{E}}\text{X}$'s elegance.

My current thought is that we should stop worrying about why $\text{T}_{\text{E}}\text{X}$ isn't more popular and simply concentrate on making its use the maximally pleasant and productive experience we can for those people who, for whatever reason, choose to use $\text{T}_{\text{E}}\text{X}$.

²¹People I know who were users of Nota Bene and Word Perfect stay with them relentlessly because they know them and don't want to learn Word. Personally, I upgrade Word as seldom as possible because I don't want to learn how Microsoft has changed it. I know people who stayed with WordStar for years after it was dying.

Acknowledgements

Karl Berry read a draft and informed me regarding some nuances. The comments of the anonymous reviewers were very helpful and reassuring. Tristan Miller's and Will Robertson's comments were also useful. David Elliott caught a couple of typos at the last minute.

Comp.text.tex is a forum for the discussion of TeX, LaTeX and other related things. TeX is a software system written by Donald Knuth to typeset text, especially text containing mathematics. LaTeX is a set of macros written in TeX, designed to simplify the typesetting of a document by allowing the user to concentrate on the content and structure of the document rather than the exact appearance of the finished product. METAFONT, also discussed here, is a program which allows the user to design their own fonts. \quad comes from TeX (i.e. from the very core), so there is no need for some special LaTeX package to allow you using it. " Vasily Korolev Mar 14 '10 at 0:07. Noram: I am a bit OCD like that. I like to know the original package and documentation for any command I use. I do the same with MSDN when I use Microsoft APIs. " Ashwin Nanjappa Mar 21 '10 at 4:43. Otherwise, if it's in the index to the LaTeX manual, it's probably defined in latex.ltx or in one of the standard class files, not in a package. Otherwise, if it's in the index to The LaTeX Companion, the page number probably tells you what package it's from. Otherwise, you could do some fancy grepping on the results of `find /usr/share/texmf -name '*.sty'`, but be prepared for a painful exercise. Or, you could ask on <http://stackoverflow.com>. LaTeX is not a word processor. It's a document preparation system. Rather than type words and then format them using drop-down menus, in LaTeX the formatting is part of the text, all of which is written in ASCII characters. At first, this seems bizarre, but after a while (not too long a while) you begin to appreciate it. LaTeX does exactly what you tell it. You can see what you are telling it; if something goes wrong (and it will) you can try to find the problem yourself, and, if you can't, you can show it to others. You can even e-mail it. Try doing that with something you did in a word processor; Well, I was using version 9 and I clicked on this, and then on that, then the pull-down menu appeared and I clicked on the default . . . then I entered 2. Sheesh.