Speaking notes for “An Introduction to TEX and Its World”

drafted by Dave Walden

December 5, 2005

Some notes about this presentation

This presentation involves two files: `show.pdf` (compiled from `show.tex`) and `notes.pdf` (compiled from `notes.tex`).

The notes include the hypothetical complete text of what I might actually say. This is my approach to creating such teaching materials. I first write it as if I was speaking it, to make it sufficiently informal and comprehensive. Then for actual presentation I write a few small notes on each notes page, since of course I can’t read the whole text during the presentation. The result is very spontaneous but fluent since I am not reading it but I already have in the back of my mind exactly what I might say, especially if I had more time.

Someone giving this presentation instead of me should read the full length text a couple of times and from this create their own notes to guide them in giving the presentation including adding and subtracting from what I have suggested and otherwise personalizing it. Someone else will also want to recompile the `show.tex` file to put his or her own name on the title page. He or she will also need to decide how to handle the fact that several of the example files comes were mine personally, e.g., “A colleague of mine wrote this letter to his sister. . . ”

My assumption is that I would hand out a hard copy of the slides and handouts for the listeners to have, look at, and make notes on as I give the presentation. When I speak, I use a slide every minute or so on average, so this might be a 30 minute presentation as currently drafted. Another presenter may want to gloss over some of the slides or parts of slides.

The presenter will have to decide how to handle the issue of how different TEX’s edit-compile-display cycle is from completely visual systems such as

Word that the listeners may be used to. If talking to Windows PC users, this presentation may be slanted more or less correctly as it is. Linux/UNIX users may (will probably?) already be used to the concept of an edit-compile-display cycle and may only need briefing on what TEX itself does.

Things that may need fixing

1. I'm just used the rudimentary slide class right now. Maybe I'll change to something better later. For now I wanted to concentrate on just getting something on the page and so I could get some feedback on the use of this slide show. There are two main files (the slide show and the speaking notes) and lots of other files that get pulled into the main slide show file as full-page PDFs.

2. Presenters will need to collect or create additional handout materials they want to go at the end. I welcome suggestions for permanent additions to the set of handouts.

Slide 0: Title page

[*I think presenters should just pronounce TEX the "right" way and wait for someone to ask why it is pronounced differently than it looks from its spelling. My answer is at the end of the notes for this slide.*]

Hello, my name is Dave Walden, and I am pleased to be here to tell you a little bit about the powerful tool known as TEX. My e-mail address is on this slide, so you can follow-up with me after this presentation today, if you want to.

Before I begin my presentation, who here has used TEX before (please raise your hands)? Who has not used it but has heard about it? Who has not heard of TEX? Thank you. I will try to adjust my presentation on the fly to make it more appropriate to this specific group.

Also, before I begin my presentation, I'd like to give a couple of ground rules. I'll welcome interruptions when you need clarification about what I am trying to convey to you, although sometimes I may ask that you wait for an answer until I get to that in the natural progression of the presentation. Also, if you think have said something factually incorrect, feel free to note it. However, I will appreciate it if you to not interrupt me to express contrary philosophical positions; let's hold off on philosophical debates until the Q&A session at the end of my presentation.

[*I think of the pronunciation of T-e-X as sort of an in-joke by its creator, a mathematician and computer scientist who I will mention in a moment. He sees the X in TEX as a form of the Greek letter chi, and he pronounces it more like "tech" than like "tex" as in Texas. His full in-joke is to pronounce it with a sort of guttural overtone — kind of like you were speaking German — Tecchhh, rhymes with blecchhhh.*]

Slide 1: What TEX is

Let's get down to business and look at what TEX is.

In many ways, TEX is a word processing system: you can type text into it, format the text a bit, and print out a nice looking document. In this sense, it provides the same function Word or Word Perfect provide.

In another sense, it is a document preparation system; it knows about the components of various kinds of document. For instance, TEX has ways for you to concisely indicate the chapters or section titles for a book, or another kind of document, and it takes care of formatting them and referencing them from a table of contents.

Many people think of TEX as a typesetting system. It has powerful capabilities for handling hyphenation, inter-word spacing, inter-character spacing, and the other picky details that a professional typesetting person handle using QuarkExpress, InDesign, or other commercial systems one might buy.

With the above mentioned capabilities, TEX is used to prepare — for press publication or network display — newsletters, journals, advertising flyers, and publications of many other types.

I won't spend much time on the history of TEX during this presentation. However, it is worth noting that it was created by a noted mathematician and computer scientist named Donald Knuth for the original purpose of typesetting his highly technical set of books on *The Art of Computer Programming*. Professor Knuth began his work in response the poor quality of typesetting that was available at the beginning of the transition away from non-computer-based typesetting methods and to computer-based methods. As Professor Knuth developed TEX he developed many innovations in computer typesetting, and he made this technology freely available to the world.

Slide 2: Intro to examples

Let me show you some examples of documents created with TEX.

First, I'll show you examples from several documents that I created using TEX. These are not very fancy, but it's the sort of thing I do as a non-expert user of TEX.

Second, I'll show you several examples by people using TEX in fancier ways.

Slide 3: Example 1 — personal letter

This is an example of a letter to my sister. How it looks doesn't really matter to me. I am just using TEX as a word processor — really, as nothing more than a typewriter.

Slide 4: Example 2 — draft technical document

Here TEX is used to draft one-page technical paper (on the subject of "What Is TEX?" — you might read it later). The document was produced using a standard style template that is available for submitting documents to the particular journal which knows about formatting the title, use of a double column format, and so on. This journal paper template leaves the page numbers open until the paper is accepted and its place in a particular journal issue is known.

[*For more one-page views of what TEX is, see*
`http://tug.org/pracjourn/2005-3/walden-whatis/`]

Slide 5: Example 3 — privately published oral history — 5 pages

Here are five pages from an oral history of my mother. My wife interviewed my mother, tape recorded the conversation, and edited the conversation into book form. I originally formatted the book using Microsoft Word, and we had it copied by Kinko's and bound by a thesis binding service.

Later, I did a little work — really very little work — to reformat the text as a TEX document, and the five example pages here are from that TEX-based version.

The first page here is the title page from the oral history.

Answer to the question "Why did you convert from Word to TEX?": I knew the TEX version was likely to have different page numbering than the Word version; while I had manually inserted the page numbers in the table of contents of the Word version, I wanted to use TEX's capability to automatically insert the correct page numbers in the table of contents. Also, I wanted to add cross-references to the page numbers of the photo pages; these were not included in the Word version, and this is easy to do with TEX. Also, TEX does a prettier job of spacing the words and letter on a page that is right and left justified on a page than Word did. I'll talk about some of this more later in this presentation

My goal here is not to make an argument for TEX over Word; I am just trying to answer the question of why I happened to choose to convert from Word.

Slide 6: photo

Here is the next page after the back of the title page.

Slide 7: table of contents

Here is the first page of the table of content.

Slide 8: chapter 1 page

Here is a typical page of text — the first page of a chapter.

Slide 9: photo page

Here is a typical page with photos.

Slide 10: Example 4 — Business letter

This is a business letter. Everything in the letter, including the parts that look like letterhead stationery, are part of a TEX style that the company has available for use with business letters. The company might have other styles available for memoranda and various other standard company documents.

Slide 11: Example 5 — Botanical garden brochure

This is a page from a brochure for a Botanical garden. It was created using TEX by Mr. Joe Hogg who is a real estate agent and volunteer guide at the botanical garden.

Slide 12: more about botanical garden brochure

Mr. Hogg has described how he created this brochure in a paper at the Internet address you see on this page. He used several standard style packages that are readily available to help with the formatting of the brochure.

Slide 13: Example 6 — Company newsletter

This example comes from Bob Kerstetter, who posted it in the "General typesetting" category of the TEX showcase (`http://www.tug.org/texshowcase/`). The developer of this newsletter doesn't claim that this is particularly pretty, but it is a practical application of TEX that was convenient for him.

Slide 14: Myriad application areas

I have shown you examples of a few application areas and directed you to lots of other examples at `http://www.tug.org/texshowcase/`. However, these are only a fraction of the areas in which TEX has or is being used, as shown on this slide. TEX is used for all kinds of documents, every possible application area, and many languages.

Now its time to dig into a little more detail about where TEX comes from, how it works, and how to use it.

Slide 15: Small example of TEX output

Here is an example of the output of a pretend one page article.

Slide 16: Source of the example document

This page is what was typed into the computer to produce the previous page of output. Read the previous page while looking at this page and the previous page side-by-side.

Keep the previous two pages in mind (or actually in front of you) as we discuss this example.

Most word processing and page layout systems these day work in a very visual manner: with the exception of line wrapping, the user does all the formatting him- or herself, using menu commands, keyboard short cuts, and so forth. TEX does not work this way: rather it is based on an older and in some ways more powerful approach and tradition — an approach of explicit commands. Using the TEX approach, you use a text editor such as Notepad on Windows PCs, SimpleText on MACs, and vi or Emacs on Linux systems to type commands into a file — commands that specify the logical structure of your document in the context of a document style you have chosen. Then you tell TEX to process the command file (such as the second example page above), and it produces an output file suitable for printing or whatever (such as the first example page above).

This way of specifying the document you are writing may seem harder than a strictly visual approach such as is used with Word. However, much of the time it takes less work to to specify a document in TEX, and the user arguably has more control over the details of how the document is printed or displayed. It may be a slightly new experience to work this way, but anyone can learn to do it; in fact, millions of unsophisticated computer users used such command-based systems back before Word Perfect and Word drove Wordstar out of the word-processing market.

I am not going to argue about whether the TEX approach is better than a more visual approach or whether a more visual approach is better. (Clearly the more visual approach is more popular with a majority of people these days.) Suffice it to say that many people find the TEX approach beneficial for at least some of their work. You will have to decide for yourself if trying TEX is worthwhile for you. (During the Q&A session at the end of this presentation, I'll be glad to talk about when I use TEX and when I use systems such as Word.)

Before we dig into more sophisticated examples, it is time to clarify some things about the TEX system.

Slide 17: Levels of TEX

[*Again, I would just pronounce LATEX however the presenter pronounces it and wait for a question to deal with the pronunciation. My answer is at the end of the notes for this slide.*]

When Donald Knuth invented TEX, he developed a very sophisticated computer program to process TEX documents and typeset them. This is called the TEX typesetting engine. It has lots of primitive commands, but trying to create a document using only native TEX primitive commands is a little too low level to be practical.

Therefore, Knuth allowed for the possibilities of what he called "macro packages" to expend the typesetting engine's capabilities. In fact, Knuth developed a macro package known as "plain TEX" that lets the user work at a somewhat more abstract level of document specification. Knuth called such expansions to the basic typesetting engine "formats."

Later, a fellow named Leslie Lamport wrote a different set of macros that included most (but not all) of the capability of plain TEX and also let the user specify documents at a considerably higher level of abstraction than plain TEX. He called these the LATEX format. More people use LATEX than any other version of TEX. The example we have been studying is actually specified using the LATEX macro package on top of the TEX typesetting engine. For LATEX Lamport provided several templates for different types of documents which are called "classes": the letter class, the article class, the report class, the book class, etc. A large number of add-ons to LATEX and its classes have also been developed by many different people to provide specific additional capabilities; these are called "styles." Also, other people have written many alternative and additional classes and styles for them.

There are also other formats and other integrations of TEX and other systems, e.g., for more complicated documents, other languages, etc.). Some of these replace LATEX on top of plain TEX in the figure, some are on top of LATEX and some connect in some other way. In any case, we will not get into them in this introductory presentation.

Suffice it to say that TEX is a very open-ended system that anyone who cares can contribute to, in small ways or large ways. Also, anyone can make their own changes as we will begin to show in a later example.

For the rest of this presentation, we will talk about LATEX, which is actually what we have been talking about all along.

By the way, this figure was specified within LATEX — no drawing program like Adobe Illustrator was used.

[*The "T-e-X" in LaTeX is pronounced in the same way as in TeX alone, however you have chosen to do that. Some people pronounce LaTeX lah-tech, some people pronounce it lay-tech, and I'm sure a lot of people pronounce it latex like the kind of paint. It doesn't matter much to me, but I'm sure it matters to some people. You are probably safer avoiding the way that sounds like a type of paint.*]

Slide 18: Some complications

In LaTeX (and TeX) some characters can't just be typed as they can with the typical visual word processor.

Lines a to c of the slide show the character sequences used to type an em-dash, en-dash, and hyphen. Em-dashes are the long dashes used to indicate a parenthetical expression. En-dashes are used in numeric intervals. Hyphens are use for hyphenation. (How many of you know how to type an em-dash or en-dash in Word?)

As shown on lines d and e, an open single or double quote is indicated using one or two instances of the grave accent key on the keyboard. A close single or double quotes is indicated using one or two single quotes.

The characters shown on line f1 are reserved to help us specify various LaTeX commands, etc., and thus these characters need to be specified in some other way. The characters on line f1 are typed as shown on line f2. There are three other typical keyboard characters (a backslash, carett, and tilde) that require special character sequences to input that we won't bother to list here.

Also, LaTeX ignores single carriage returns, unlike the typical word processor where an explicit carriage return (Enter key) indicates the end of a paragraph. Instead, LaTeX uses an extra blank line (e.g., double Enter) to indicate the end of a paragraph. Look at where the paragraphs are on the first page of the two-page example above and where there are blank lines on the second page.

LaTeX also tries to be pretty smart about adding extra horizontal space after periods, on the assumption that a sentence has ended. However, this requires a little bit of special knowledge to avoid extra space where you don't want it, e.g., after the "r" in "Mr. Jones".

I'll also note here that LaTeX has a very sophisticated hyphenation algorithm. See the full page example again.

None of these so-called complications is actually very hard, and one catches on within an hour or two of use of LaTeX.

Slide 19: Example 7 — bigger LaTeX example — starting template

Now let's look at a more significant example of the use of LaTeX.

This time I am going to use LaTeX's standard letter style rather than the article style I used before.

Suppose I have a letter template set up as shown in this slide. The letter template includes my return address, name for the signature line, a place for the address to which the letter is being sent, a place for the salutation, and my standard closing. I could have other standard parts of the letter such as enclosures and a cc list.

Slide 20: Example 8 — bigger LaTeX example — sample letter output

Here is a letter created by use of this template.

Slide 21: Example 9 — bigger LaTeX example — what the user typed

And here is what the user types to get the previous page. Look over this page carefully and keep glancing at the previous page to see what was produced from this page.

At the top of this page we just see the template items with the address and salutation fields filled in.

Next we have the letter content.

One big new thing about this example is the use of different type fonts such as italics, typewriter, and bold. Think about how you specify font changes (e.g., to italics) in your visual word processor: you probably do something like select a sequence of words, pull down a format menu, select the font option, and then click the box for italics (or maybe you use the short cut of selecting the text and typing control-i). This font information is stored invisibly with the characters in some way. With LaTeX you just specify the characters you want in a different font by embedding the characters in a command like `\textbf{ }` for bold face, `\texttt{ }` for typewriter font, or `\textit{ }` for italics.

I have also put backslashes around the word TEX so that it is typeset like its inventor wants it typeset.

Another big new thing is the way I have created an enumerated list.

Before I leave this page, let me come back to the instances of *TUGboat* in italics. Suppose I had a big document with many instances of this throughout the document as well as many instances of other words in italics, and I decided that I wanted to change *TUGboat* in italics to `TUGboat` in a typewriter font. I could just use my editor to do a "Replace-all" of the character sequence backslash, t-e-x-t-i-t, open brace, TUGboat, close brace with the character sequence backslash, t-e-x-t-t-t, open brace, TUGboat, close brace. I don't know how I would make such a change in Word without a lot more work.

Slide 22: Example 10 — bigger LATEX example — revised user input

I have made three changes to this version of the TEX input.

1. In the first line of the page I changed the size of all of the text in the output page to make it a little bigger. I did this by adding the 12 point indication in square brackets. Take a look at the next page to see the result.

2. Continuing on the revised input page, I'm going to hint at one of LATEX's most powerful features — the capability for users to create new commands — essentially, to program TEX. Notice that in the previous version of this input I had to type out `\textit{TUBboat}` several times. On the second line from the top of the page, I have defined new command called backslash-T-B that is equivalent to `\textit{TUBboat}`, and in the content of the letter I have used my new abbreviation. Look at the next page, and you will see that backslash-T-B still produces the journal name in italics. Of course, this abbreviation would be a lot more useful in a lot longer document.

3. Finally, suppose I'm not sure of the order of the items in the list, and I would prefer to cross-reference them as I do in the last line of the

letter in a way that automatically changed the cross-reference number if I changed the list order. Of course, this would be much more useful if the cross-reference was from many pages away, as it might be in a long report. In the second item of the list, I use LaTeX's capability to define a symbolic label. Then, in the fourth line from the bottom of the example, I use LaTeX's capability to reference a symbolic label which results in the number of the item being substituted for the reference; notice on the next page that this still turns into the number 2. However, if I exchanged positions of the second and third list items, the last sentence of the letter would still refer to the correct item in the list, now the number 3.

Slide 23: Example 11 — bigger LaTeX example — revised output

And here is the output of those changes.

Since this presentation is about what LaTeX is rather than how to actually use LaTeX in detail, I'll stop giving how-to examples now. I hope you comprehend the non-visual way things are specified in LaTeX, see that it has a relatively simple logic of its own, and perhaps you can imagine cases where the LaTeX approach is more powerful or easier than a more visual approach.

Slide 24: Where TEX is especially useful

This slide list some of the areas where TEX has proved to be particularly useful:

- typesetting
- math
- codifying style of document to application
- large documents
- languages
- extensibility
- system independence
- can use the editor of your choice
- targeting different output systems

Typesetting. TEX does an excellent job of laying out characters, words, and paragraphs on the lines of a page. It relieves the user of most of having to deal with this to get high quality output, although sometimes the user much make a few final tweaks to the TEX commands to finalize the output.

TEX is generally regarded at the preeminent typesetting system for typesetting mathematics. In fact, it is the standard system used by mathematicians, economists, and others who write a lot of mathematics.

Slide 25: Math mode

Math. By just bracketing math symbols by dollar signs, TEX turns the symbols into good looking math: a math font is used, superscripting is trivial, etc.

By bracketing the math symbols with double dollar signs, TEX knows enough to center the math equation or term as it might be in a math book or paper. TEX also has all the mechanisms for automatically numbering such math displays and cross-referencing them.

Now you see why you have to type a backslash before a dollar sign when you actually want to show a dollar sign rather than include math terms.

Slide 26: A page of math

Here is an example of lots of typesetting of math submitted by Martin Jansche to the TEX showcase.

Slide 27: Where TEX is especially useful

Codifying document style. We have already seen examples of codifying the style of a document to the article and letter applications. The user works in logical terms such as `\title`, `\author`, `\section`, and `\footnote` TEX takes care of handling the visual interpretation of the logical elements.

For instance, in this presentation using the `slides` class, I am using logical elements for defining the beginning and end of a slide, and TEX is choosing the way to display the slide.

Large documents. TEX is exceptionally useful for dealing with large documents, such as a many page report or a book. It does automatic numbering of chapters, sections, figures, tables, footnotes, citations, and so forth. Any of these can be cross-referenced. When, for instance, the position of a figure is changed or a figure is added or dropped, all of the cross-references are automatically renumbered.

When I write a book, I typically make each chapter into a separate file. TEX has features that allows the user to declare the current files (chapters) of interest, and only those are processed. This makes it much easier to work within one chapter, for example for searches and replaces.

If I later decide to change the visual format of chapter headings, I just change TEX's definition of a chapter heading, and all of the chapter headings use the new definition. There is no need to go through the whole book manually changing the start of every chapter.

TEX has a very flexible capability for handling bibliographic citations, and you can chose among lots of formats or make up your own. In fact, the formats used for many journals are already available in the TEX library of capabilities.

Languages. I have already mentioned that TEX has the capability to handle many alphabets. It also can typeset right-to-left, left-to-right, or a mixture of the two.

Extensible. TEX was built to be extensible. Average users or experts can define new commands that are shortcuts for frequently typed text or abstractions of frequently used sequences of commands. Experts create new packages to typeset a particular type of application, e.g., a chess book.

Unfortunately, this is where some complexity in trying to use TEX comes form. Because TEX is so expandable, users try to change it in ways they would never try to change Word, but these changes can sometimes be a little tricky. The power is there and we want to use it, and then we sometimes complain about why it is not a little easier to do.

Operating system independent. TEX runs under Linux/UNIX, on MACs, on Microsoft Windows machines, and on just about every other kind of system. Thus, users can exchange and collaborate on documents regardless of which computer operating system they favor.

Also, because TEX documents are created in plain text files without lots of hidden, proprietary markup, documents can be exchanged as part of email messages themselves rather than as attachments. This is a nice feature in today's world where viruses and worms sometime lurk in attachments.

Editor independent. Unlike Word or QuarkExpress where the editor is part and parcel of the word processing or layout program, with TEX one can use any editor that is available on your computer. Thus, you can choose to stay with an editor with fairly weak capabilities, you can choose an editor with powerful capabilities, or you can choose an editor or whole development environment that is highly oriented toward TEX.

Flexible output. At the end of this list, but not necessarily at the end of TEX's areas of high performance, with a few command changes TEX can produce output for professional printing, output to a home printer, output for the web, and so forth.

Slide 28: Getting started with TEX

- Obtaining the TEX system

  There are lots of way to obtain TEX especially since it runs on so many types of computers and operating systems.

  Each year the TEX Users Group provides a CD or DVD with the complete TEX system to its members. Anyone can download the complete TEX system for free. TEX is also available from many commercial vendors. Pointers to various sources of TEX are included in one of the handouts at the end of this presentation. If you are a Linux or UNIX user, TEX may already be on your system.

  You will also need to decide what editor to use — one that is already available on your computer or another editor that is perhaps more oriented to TEX. Linux/UNIX users may just keep using their existing editors, e.g., **vi** or Emacs. Many windows users WinEdt (which is already inexpensive and available to TUG members with an additional discount).

  There are also many editors (commercial and otherwise) that make TEXuse be more visually oriented, that is, more like using Word.

- Books, manuals and tutorials

  There is a vast body of freely and commercially available literature on TEX. Some of these are listed on a handout page.

- Obtaining consulting help

  There is also a vast network of other resources for getting help with using TEX. TUG has the **texhax** discussion group, and TUG members help answer questions on the **comp.text.tex** discussion group available by clicking on "Groups" on the Google home page. Many TUG members also help maintain CTAN — the Comprehensive TEX Archive Network. Also, the TEX FAQ is a good place to start with any question about TEX.

- Joining a users' group

  I wish you would join the TEX Users Group, and its activities are described in the next slide. There are also many national (or language-based) user groups, for example, in Germany or China.

Slide 29: TEX Users Group (TUG) history and function

TUG was established to promote the expansion and well-being of TEX and to support TEX users.

TUG was originally substantially a developer's group — all of the people trying to understand Knuth's then new TEX system and implement it on different systems met and communicated through TUG.

That was many years ago. Now TEX is readily available on most computers and many of the original workers have gone on to other things. Today TUG mainly serves TEX users, although it also has a modest program to continue to push the development of TEX.

The main tangible product of TUG is its journal *TUGboat*, which is now over twenty-five years old. *TUBboat* publishes a mix of introductory, intermediate, and expert material relating to TEX. It also includes some news about the TEX world.

Each year, TUG also sends out CD-ROMs and a DVD that contain an up-to-date version of the TEX system. TEX changes only slowly, and there is no need to upgrade every year, but there are always a few new features that it would be nice to have. The CD-ROMs and DVD also contain an enormous library of add-ons to TEX, and it is very handy to get the latest version of this periodically.

TUG holds an annual meeting and conference that rotates to different locations in the U.S. and around the world. This is a good place to meet other TEX users, take a tutorial course, and hear about the state-of-the-art of TEX use.

The TUG web site lists many other resources and activities that may be of use to different groups of TEX users. Take a look at `tug.org`.

To do its work and pay the accompanying expenses, TUG needs to collect annual fees from members, even though most TEX activities are primarily staffed by volunteers. Students, seniors, and several other groups can get discounted memberships. This presentation was developed by a TUG volunteer.

Slide 30: Thank you

Thank you for your time and attention.

[Wait a couple of seconds in case there is any applause.]

Slide 31: Q&A period

I'll be glad to try to answer a few questions. But since this is only an introduction and not a tutorial, please mostly stick to questions about TeX rather than how-to-in-TeX questions.

Also, look in the back of your notebook for some additional useful information

Slide 32: List of handouts

- list of useful URLs (CTAN, TUG, comp.text.tex, texhax, . . .)