

# Practical T<sub>E</sub>X 2006 — program and information

<b>Tuesday July 25– Friday July 28</b>	9 am–5 pm	<b>L<sup>A</sup>T<sub>E</sub>X workshop</b> <i>led by Sue DeMeritt &amp; Cheryl Ponchin</i>	
	8–9 am	<i>registration</i> (on Tuesday, at Rutgers)	
	10:15–10:30 am	<i>break</i>	
	12–1 pm	<i>lunch</i>	
	3–3:15 pm	<i>break</i>	
<b>Saturday July 29</b>	5–7 pm	<i>registration &amp; reception</i> (at the Clarion hotel, Windsor Ballroom)	
<b>Sunday July 30</b>	8–9 am	<i>registration</i> (at Rutgers)	
	9 am	Karl Berry, T <sub>E</sub> X Users Group	<i>Welcome &amp; introductions</i>
	9:20 am	Barbara Beeton, AMS & TUG	keynote address: <i>How to create a T<sub>E</sub>X journal: A personal journey</i>
	10:20 am	<i>break</i>	
	10:30 am	Peter Flom, NDRI	<i>L<sup>A</sup>T<sub>E</sub>X for social scientists and other people who think they don't need it</i>
	11:10 am	Jim Hefferon, St. Michael's College	<i>L<sup>A</sup>T<sub>E</sub>X resources</i>
	11:50 am	Boris Veytsman, George Mason Univ. & AES ITT Industries	<i>Design of presentations: Notes on principles and T<sub>E</sub>X implementation</i>
	12:30 pm	<i>lunch</i>	
	1:40 pm	Alan Wetmore, US Army	<i>T<sub>E</sub>X and after dinner speaking</i>
	2:20 pm	Steve Peter, Beech Stave Press	<i>Fonts, typefaces, glyphs &amp; sorts</i>
	3 pm	<i>break</i>	
	3:10 pm	Klaus Höppner, DANTE e.V. & TUG	<i>Creation of a PostScript Type 1 logo font with MetaType1</i>
	3:50 pm	William Adams, Mechanicsburg, PA	<i>TypeSpec v2: Typesetting font specimens</i>
	4:30 pm	q & a, Birds of a Feather (see following page)	
	<b>Monday July 31</b>	9 am	Ned Hummel, University of Nebraska
9:40 am		Francesco Costanzo & Gary L. Gray, Penn State University	<i>Examples of readily implementable, but "fancy", page layout elements and organizational structures for a book</i>
10:20 am		<i>break</i>	
10:30 am		Federico Garcia, Univ. of Pittsburgh	<i>Capabilities of PDF interactivity</i>
11:10 am		Boris Veytsman, GMU & AES ITT	<i>Automatic report generation with Web, T<sub>E</sub>X and SQL</i>
11:50 am		Kaveh Bazargan, River Valley Technologies	<i>Removing vertical stretch: Mimicking traditional typesetting with T<sub>E</sub>X</i>
12:30 pm		<i>lunch</i>	
1:40 pm		David Walden, E. Sandwich, MA	<i>A lifetime as an amateur compositor</i>
2:20 pm		Troy Henderson, US Military Academy	<i>Creating high-quality technical graphics with MetaPost</i>
3 pm		<i>break</i>	
3:10 pm		Andrew Mertz & William Slough, Eastern Illinois University	<i>Graphics with PGF and TikZ</i>
3:50 pm		Jon Breitenbucher, College of Wooster	<i>L<sup>A</sup>T<sub>E</sub>X at a liberal arts college</i>
4:30 pm		q & a, TUG meeting	
<b>Tuesday August 1</b>	9 am	Peter Flynn, Silmaril Consultants	<i>Rolling your own document class</i>
	9:40 am	Federico Garcia, Univ. of Pittsburgh	<i>L<sup>A</sup>T<sub>E</sub>X and the different bibliography styles</i>
	10:20 am	<i>break</i>	
	10:30 am	Boris Veytsman, GMU & AES ITT	<i>Drawing medical pedigree trees with T<sub>E</sub>X and PStricks</i>
	11:10 am	Elizabeth Dearborn, Buffalo, NY	<i>T<sub>E</sub>X and medicine</i>
	11:50 am	Bob Neveln & Bob Alps, Widener Univ.	<i>Writing and checking complete proofs in T<sub>E</sub>X</i>
	12:30 pm	<i>lunch</i>	
	1:40 pm	Stephen Moye, AMS	<i>A wayward wayfarer's way to T<sub>E</sub>X</i>
	2:20 pm	Steve Peter, Beech Stave Press	<i>Introduction to memoir</i>
	3 pm	<i>break</i>	
	3:10 pm	panel: Barbara Beeton, Jim Hefferon, Mirko Janc, Jonathan Kew	moderator: <i>David Walden</i>
	≈ 4 pm	<i>end</i>	
	7 pm	<i>banquet</i> (at the Clarion hotel, Garden Room)	

## Conference logistics

- The initial registration and reception is on Saturday, 29 July, 5 pm to 7 pm at the Clarion Hotel, in the Windsor Ballroom. Snacks and nonalcoholic beverages will be served. Please check in at the registration table, pick up your name tag, conference booklet and other items, and let us know if you intend to use the hotel shuttle to and from Rutgers during the conference. If you're not able to attend the Saturday reception, you can register Sunday morning from 8 am–9 am outside the conference room at Rutgers.
- The conference location at Rutgers is in the first floor lecture hall of the CoRE building, Busch Campus, 96 Frelinghuysen Rd., Piscataway, NJ 08854-8018. If you are driving, please park in lot number 64. It's located just outside the CoRE building. You will need a parking permit, which we've sent in email. We'll also have copies available at registration, and during the conference.
- The Clarion Hotel will provide a shuttle (capacity 14) to transport attendees to and from the conference. The shuttle will depart the Clarion at 8:15 am daily (Sun, Mon, Tues), and pick up at the CoRE building at approximately 6 pm Sunday and Monday, and at 4:30 pm Tuesday. If you have a car, please consider sharing the ride, as it may be difficult to reserve a second hotel shuttle.
- Morning and afternoon breaks will be held just outside of the lecture hall. Lunch will be served in the 4th Floor Lounge, Room 401, of the CoRE building.

Local area information is on the conference web site, <http://tug.org/practicaltex2006/jersey.html>. Also, in this booklet we've included a two-page note about some area attractions especially for technical folks, reprinted by kind permission of IEEE.

## Birds of a Feather

We are continuing the “Birds of a Feather” (BOF) tradition at TUG conferences. The idea is to gather in informal groups to discuss a particular issue or topic(s). This will start Sunday after the regular session. The meetings can take place in the lounge, office, outside, or any convenient location.

Topics so far:

- Customizing TeXShop: hands-on workshop (bring your laptops) led by Herb Schulz.
- Trial run of a slideshow introducing T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X, led by Dave Walden.
- Meeting journal guidelines for PDF with T<sub>E</sub>X.
- Using T<sub>E</sub>X and friends for professional publishing.
- Maintaining a multi-platform and/or multi-user T<sub>E</sub>X installation, based on T<sub>E</sub>X Live or another distribution.

More topic suggestions are welcome at any time. There will be a signup sheet at registration. Following are summaries for two of the topics above.

**Customizing TeXShop**, Herbert Schulz. TeXShop is one of the most popular “front ends” to T<sub>E</sub>X under Mac OS X. While most people are happy with TeXShop as delivered there are several ways, ranging from simple to complex, that it may be customized to satisfy the taste of each individual user. This is a short (≈2hr) workshop on how to customize TeXShop to: i) make writing L<sup>A</sup>T<sub>E</sub>X more convenient and less error prone; and ii) do things *your way*.

**Draft Slide Show: An Introduction to T<sub>E</sub>X and Its World**, David Walden. I have drafted a half hour slide show introduction to T<sub>E</sub>X (really to L<sup>A</sup>T<sub>E</sub>X) with complete speaking notes, and I would like to do a trial run of this at the conference. I will be happy to provide advance electronic copies of both the slides and the notes to anyone who wants to view my trial run. I will also be happy to take oral or email feedback and to discuss with viewers how to improve this promotional tool and make it available to anyone who wants to use it.

## TUG members meeting

After the regular session on Monday, we will hold a TUG user group meeting for anyone else interested. Several TUG board members will be present at the conference: Barbara Beeton, Karl Berry, Sue DeMeritt, Steve Grathwohl, Jim Hefferon, Klaus Höppner, Steve Peter, Cheryl Ponchin, and Dave Walden, as well as TUG's executive director, Robin Laakso. We will report on TUG's current status and future outlook.

More importantly, we invite discussion of any TUG-related business at this time: ideas for outreach to additional communities, additional initiatives TUG might undertake, existing projects which TUG might support, or anything else. Hope to see you there.

## Banquet & soapbox

The conference banquet will be held at the Clarion Hotel, Garden Room, at 7 pm on Tuesday, August 1.

Thanks to Peter Flynn for suggesting a 66–99 second soapbox at the banquet, where anyone can speak for a minimum of 66 seconds and a maximum of 99 seconds on a T<sub>E</sub>X-related topic:

We will have a few door prizes at the banquet (as usual). If you haven't signed up for the banquet, it's not too late. Please just let us know by phone (+1 503-223-9994) or email ([office@tug.org](mailto:office@tug.org)).

- You can report a success, gripe about a problem, lament a failure, share an insight, ask a question, or explain a solution.
- No intros, no questions, no hacking on earlier speakers; just you, the mike, and the audience . . .
- . . . well, and a moderator with a timer who will cut you off when your 99 seconds are up.
- No slides, overheads, whiteboards, blackboards, flipcharts, chalk, markers, or other props.
- Come prepared or make it up on the spur of the moment—no experience necessary.

## Excursion to New York City

On Wednesday, August 2, for anyone interested, we've organized a day trip into New York City.

We'll meet in the lobby of the Clarion Hotel at 8:15 am, take the hotel shuttle to the Edison train station (5–10 minute drive from the hotel) and catch the 9:11 am train to NYC ([http://www.njtransit.com/sf\\_tr\\_timetables.shtml](http://www.njtransit.com/sf_tr_timetables.shtml)).

Round trip ticket cost is \$15.50; we'll purchase our tickets at the Edison train station. The train is scheduled to arrive in Penn Station in the city at about 10 am. From Penn Station we will walk to the newly renovated Morgan Library and Museum (<http://www.themorgan.org>), hopefully arriving there around the time they open at 10:30 am for a self-guided tour. Thanks to Peter Flom for the great suggestion.

We can have lunch at the Morgan Cafe (<http://www.themorgan.org/visit/menu.asp?id=6>), or of course you can choose to eat lunch elsewhere. Those wanting to continue with the group should meet at the Morgan Library and Museum at 1 pm.

After lunch, some of us (Robin, Karl) plan to see the musical *The Wedding Singer*. Lots of other shows are available, of course. All the Wednesday matinees for which we have discount coupons are on the Schooltix web site (<http://www.schooltix.com/shows.html>).

For those who might prefer a different sort of outing, Steve Peter has offered to take an expedition to the Strand, a huge new and used bookstore (<http://www.artbook.com/strandbookstore.html>). It's a nice walk (or subway or cab; it's about a mile and a quarter) through midtown to Union Square.

## L<sup>A</sup>T<sub>E</sub>X workshop: July 25–28, 2006

For Practical T<sub>E</sub>X 2006, we are offering a dedicated hands-on workshop before the conference. To allow sufficient time to cover a significant portion of L<sup>A</sup>T<sub>E</sub>X, the workshop will be the four days preceding the conference, from July 25 through July 28 (Tuesday–Friday).

The workshop will take place in the Hill Center for Mathematical Sciences, Room 248. Computers will be provided.

Attendees may also bring specific problems and, time permitting, the instructors will attempt to work with you on them.

Curriculum outline:

- Day 1 (Tuesday, July 25): Basic document classes (article, report, etc.), sectioning, itemizing, footnotes, theorems.
- Day 2 (Wednesday, July 26): Tables, figures, simple graphics, simple mathematics.
- Day 3 (Thursday, July 27): Labeling, referencing, more graphics, more complicated mathematics.
- Day 4 (Friday, July 28): Macro writing, bibliographies; general q&a and personal project time.

## **TypeSpec v.2: Typesetting font specimens**

*William Adams*

Stephen Moye’s plain  $\TeX$  TypeSpec macros converted to work as a  $\LaTeX$  document class and organized into macros will be shown, including a number of new layouts. Additional ideas for specimen layouts will be solicited, and some will be constructed interactively during the presentation.

## **Removing vertical stretch: Mimicking traditional typesetting with $\TeX$**

*Kaveh Bazargan*

One of  $\TeX$ ’s advantages over traditional typesetting systems is the mechanism to stretch horizontal and vertical glue as needed, in order to aid paragraph building and pagination. But all  $\TeX$  operators involved in day-to-day page make-up know that this inbuilt intelligence is often ‘too clever’ and frustrating for the user.  $\TeX$  will not do what you want it to do, and only over many years can operators gain the knowledge that allows them to make just the right change to the source code in order to coerce  $\TeX$  to produce the desired result.

Recently we have been experimenting with removing vertical stretchability, with promising results. Our approach is to round off the height of all vertical material, including floats and displayed equations, to be an integral number of the leading of the main text. One advantage is that this allows true ‘grid’ setting in double column text. A live demonstration will be presented, showing our latest results.

## **How to create a $\TeX$ journal: A personal journey**

*Barbara Beeton*

When TUG was first formed, the Internet wasn’t generally available; the logical channel for communication with and among TUG’s members was on paper. So *TUGboat* came into being.

As  $\TeX$  has matured, the needs of the community have evolved, but paper is still a logical medium for showcasing a typesetting tool.

This talk will introduce high- and low-lights in the history of *TUGboat*, some reasons for choosing its particular format and mode of presentation, several experiments, and lots of my personal experiences as editor.

## **Some Examples of readily implementable, but “fancy”, page layout elements and organizational structures for a book**

*Francesco Costanzo, Gary L. Gray*

We will present a number of examples of some page layout elements we have created for two engineering textbooks we are writing. These will include:

- colored banners behind heading text (on following page, see fig. 1, item A) and colored backgrounds that don’t fill the entire page (fig. 1, left),
- colored heading text and text separators with widow and orphan control (fig. 1, item B)
- equations within colored boxes with separate colored banners that identify their original usage (fig. 1, item C), and
- marginal notes that have colored text and backgrounds (fig. 1, item D),
- separate numbering schemes for figures, tables, and equations that depend on their context (fig. 1, left).

We will provide the examples of the usage of each of these items outside the context of our manuscript and will also provide detailed explanations as well as the source code. Additional layout elements will also be presented.

## **$\LaTeX$ for social scientists and other people who think they don’t need it**

*Peter Flom*

I will discuss some ways that some people who don’t use  $\LaTeX$  (or  $\ConTeXt$  or other  $\TeX$  packages) might be encouraged to use it. I will concentrate on the case of social scientists, either in academia or in other research settings. I will also include a list of questions that I still have, or possible improvements, that might increase the utility of  $\LaTeX$  for this group.

## **Rolling your own document class: Using $\LaTeX$ to keep away from the Dark Side**

*Peter Flynn*

$\LaTeX$  users who want to package their macro application as a document class file are often put off by the complexity of the sample classes in the standard distribution. This paper describes what the code in the article files does and suggests solutions to some of the popular requirements.

## **$\LaTeX$ and the different bibliography styles**

*Federico Garcia*

The myriad of different styles for bibliography and reference layouts can be, in the main, classified into three main “families”: the ‘label’ family, where references are denoted with label, usually in [square parenthesis]; the ‘author-year’ family (Adorno, 1978); and the ‘footnote’ family. Although there are arguments for each of those families, the choice between them is, in the last analysis, decided by tradition: different disciplines have come to adopt different styles, and new generations of authors will naturally follow the uses of their predecessors (which are also enforced institutionally, for example with journal guidelines).

L<sup>A</sup>T<sub>E</sub>X itself (with B<sup>I</sup>B<sub>T</sub>E<sub>X</sub>) is designed toward the label family. Some packages, like cite, provide extra utilities in that family. The other two families are reflected in the L<sup>A</sup>T<sub>E</sub>X world by special packages. The author-year family is well illustrated by packages such as harvard, natbib, and achicago. Footnote-style references are implemented by the package opcit. A basic description of these packages follows. I will spend relatively longer with opcit, which is comparatively recent (2002) and the one I know best. A new release of it, moreover, is planned for the summer.

## Hypertext techniques with PDF

*Federico Garcia*

With the proliferation of electronic publishing, the need of L<sup>A</sup>T<sub>E</sub>X users to master techniques of PDF interactivity is growing. This presentation discusses some of the immediate additions to ‘regular’ typesetting in the electronic environment, mainly the hypertext possibilities of hyperlinks and bookmarks.

Most of these things have already been implemented successfully in L<sup>A</sup>T<sub>E</sub>X, through the wonderful hyperref package. The first part of the presentation is devoted to illustrating these possibilities. Some ‘primitive’ commands of pdf<sub>T</sub>E<sub>X</sub> are referred to too.

There is an issue, however, that is not easy to solve at the author level: how to incorporate the entries in the lists of tables and of figures in their own sets of bookmarks. A solution for this, implemented by the document class pittetd in 2003, is shown here.

## L<sup>A</sup>T<sub>E</sub>X resources

*Jim Hefferon*

One of the main practical advantages of using T<sub>E</sub>X via L<sup>A</sup>T<sub>E</sub>X is the existence of a great many, widely available, resources. These include add-on packages, books, articles, web pages, and a long history of question and answer discussions. We will discuss some of the main resources and how to use them, including some of most important add-on packages.

## Using MetaPost for creating high-quality technical graphics

*Troy Henderson*

MetaPost is a powerful graphics language (by John Hobby) based on Donald Knuth’s METAFONT with high quality PostScript output. An outstanding feature of MetaPost is that typeset fonts in the output graphics are consistent with those in T<sub>E</sub>X-based documents. In this talk we discuss a brief history of MetaPost as well as reasons for wanting to use it. We will also discuss some of

the basic syntax of MetaPost coupled with several examples. Finally, we show how to include these graphics into L<sup>A</sup>T<sub>E</sub>X documents as well as convert them to more web-friendly formats such as Portable Network Graphics (PNG) or Scalable Vector Graphics (SVG).

## Creation of a PostScript Type 1 logo font with MetaType1

*Klaus Höppner*

MetaType1 is a tool created by Bogusław Jackowski, Janusz Nowacki, and Piotr Strzelczyk for creation of PostScript Type 1 fonts. It uses MetaPost, t1utils and some AWK scripts to create a MetaPost font source with some special macros.

MetaPost was used to create the Latin Modern fonts, which are derived from Computer Modern fonts but include a lot of characters with accents. It is part of most modern T<sub>E</sub>X distributions. Additionally some original fonts, notably Iwona and Kurier, were created by the developers of MetaType1.

I came into touch with MetaPost when I wanted to convert an existing logo font from METAFONT to PostScript Type 1. Unfortunately there doesn’t exist a tutorial or cookbook for using MetaType1. So I started to play with the example fonts supplied as part of MetaType1 and to read the comments in the source. This tutorial will show a simple example and the lessons I learnt.

## Common macro pitfalls and how to avoid them

*Ned W. Hummel*

In the process of learning L<sup>A</sup>T<sub>E</sub>X there are a number of common pitfalls that many of us fall into at some point.

For most of us we encounter these pitfalls when writing macros for the first time. One of the great advantages of L<sup>A</sup>T<sub>E</sub>X is the ability to use macros to logically markup our document. Unfortunately, a number of us tend not to apply that same logical markup philosophy when writing macros.

We will consider a number of examples and discuss ways to re-write them using a logical markup philosophy.

## Graphics with PGF and TikZ

*Andrew Mertz, William Slough*

Beautiful and expressive documents often require beautiful and expressive graphics. PGF (portable graphics format), and its front-end TikZ walks a thin line between power, portability and usability, to give a T<sub>E</sub>X-like approach to graphics. While PGF and TikZ are extensively documented, first-time users may prefer learning about these packages using a collection of graduated examples. The examples presented here cover a wide spectrum of use, and provide a starting point for exploration.

## Fonts, typefaces, glyphs & sorts

*Steve Peter*

This presentation focuses on the general characteristics and usages of typefaces, without specific reference to T<sub>E</sub>X. I will begin by covering the history of printing technologies and offering an overview of some useful classification schemes for typefaces. I then turn to a practical discussion of selecting the right typeface for the right job, with a nod to using T<sub>E</sub>X to its fullest.

## Introduction to memoir

*Steve Peter*

This presentation serves as a gentle introduction to Peter Wilson's memoir class, an alternative to the standard L<sup>A</sup>T<sub>E</sub>X classes. Memoir is quite flexible, and makes it easy to create beautiful book, article, and report designs, without having to search for, install, and load numerous third-party packages.

## Design of presentations: Notes on principles and T<sub>E</sub>X implementation

*Boris Veytsman*

There are many T<sub>E</sub>X packages available for creating presentations. Mostly they imitate the ubiquitous style of a certain tool, striving to produce PowerPoint-like slides, hopefully with better typographical execution.

In this talk the principles of good design for presentations are considered. We discuss the problems with the common design of presentations as well as the famous proposition by Tufte to avoid slides at all. We try to formulate the principles of good presentation design and discuss T<sub>E</sub>X implementations from this point of view.

The discussion is based on the author's experience in making slides for talks, lectures and training sessions.

## Automatic report generation with Web, T<sub>E</sub>X, and SQL

*Boris Veytsman, Maria Shmylevich*

One of the most time-consuming task of a manager for a federal contractor is the creation of reports: weekly, monthly, quarterly and yearly as well as special reports at the end of project or any given date. Such reports are usually made by copying and pasting the daily reports of subordinates.

The system described in this talk makes these reports automatically. The members of project team file their daily work results using a Web interface. These entries are kept in a SQL database. The report generation utility is launched through Web interface. It creates a T<sub>E</sub>X file by selecting the data relevant to the given set of contracts and tasks, employees, time periods, etc., and collating the individual reports. The result is then put through pdfT<sub>E</sub>X or latex2html or latex2rtf to create either a PDF report or editable (e.g., in Microsoft Word) file.

## Drawing medical pedigree trees with T<sub>E</sub>X and PStricks

*Boris Veytsman, Leila Akhmedeyeva*

Medical pedigrees look like genealogical trees, but also have certain interesting features. Usually they are drawn by hand by medical geneticists. This is a cumbersome and time-consuming process. Freely available programs for drawing genealogies are not fully suitable for this task because of the special format of medical pedigrees.

We discuss a package for drawing pedigrees based on PStricks. The information is input by geneticists in a spreadsheet; a Perl program extracts it and calls T<sub>E</sub>X to produce the final output.

## T<sub>E</sub>X and after dinner speaking

*Alan Wetmore*

I will discuss a somewhat novel use for T<sub>E</sub>X, preparing an after dinner speech for a scientific conference. My experience some years ago required me to prepare, at quite short notice when a scheduled dignitary was forced to cancel, an entertaining diversion for the attendees of a conference banquet. Inspired by the then-current popularity of *Who wants to be a Millionaire?*, and *The Weakest Link*, I produced a domain-specific trivia "contest" based on some frenzied Internet sleuthing. Formatted using `pdfscreen` and `pdfLATEX` I produced an attractive presentation for the audience. In the process I learned a little about various T<sub>E</sub>X's presentation capabilities. Some examples will be given.

# A Busman's Holiday in New Jersey

Michael N. Geselowitz  
IEEE History Center

Many visitors to the IEEE Operations Center in Piscataway, New Jersey, think of it as a stopping point between Philadelphia and New York City. Philadelphia and New York are both major metropolitan areas and also meccas for tourists. Not surprisingly, given their political, economic, and industrial history, both cities are full of attractions for the tourist with a particular interest in the history of technology—from the Franklin Institute and the ENIAC display at the University of Pennsylvania in Philadelphia to Christie's auction house (site of the recent "Origins of Cyberspace" sale) and the Transit Museum in New York.

However, before racing from one city to the other, visitors might wish to consider the role of central New Jersey in technological history, and to seek out some lesser-known gems in smaller towns on the tourist trail. From the telegraph to tetrahertz switches, New Jersey corporations, entrepreneurs, government laboratories, and universities have been at the forefront of electrical, electronic, and computer technology.

As an example, the IEEE Operations Center sits on land that formerly belonged to AT&T. Telcordia corporate headquarters sits across the street. There is also the Liberty Science Center in Jersey City, which is a science and technology museum of national import.

Perhaps the most exciting attractions to interested tourists are the plethora of smaller museums that have preserved important aspects of New Jersey's technological heritage—which turns out to be the world's technological heritage—and made them accessible to the public while maintaining a local flavor. Following is a sample of what is available merely by making a loop that is less than 175 miles.

## Historic Speedwell

Beginning in the early 19th century, New Jersey was truly a global center of what was then high tech. Samuel F.B. Morse did most of the work on his telegraph system at the home of his partner, Alfred Vail in Morristown, just 30 miles north of Piscataway. The Vail's Homestead Farm was also the site of the family's ironworks (a previous

owner had made the machinery for the S.S. *Savannah*, the first steamship to cross the Atlantic). The farm, now known as Historic Speedwell, is a US National Historic Site. It's open to the public and includes displays on iron-working, 19th-century New Jersey life, and the invention of the telegraph.

## Morris Museum

Also in Morristown, only three miles from Speedwell, is the Morris Museum. Although mainly a museum of local history, art, and science, it is perhaps the best hidden treasure of central New Jersey, because it happens to house the Murtoth D. Guinness Collection. Readers of *Annals* will be aware that there's an interesting tradition in computer history known as *automata*.

There were classical antecedents, a resurgence in the 18th century, and by the 19th century, automata became commodities for the amusement of the upper and upper-middle classes in Europe and the US. In the 20th century, Murtoth D. Guinness, one of the heirs to the Guinness brewery fortune, accumulated one of the largest collections of these robotic figures—nearly 700! In 2004, the collection came into possession of the Morris Museum. Currently, about 60 masterpieces of the collection are on display while the museum constructs a new wing that will house the new exhibition space plus open storage that will show the entire collection.

## Edison National Historic Site

Less than 20 miles east of Morristown is the Edison National Historic Site in West Orange. Thomas Edison made his first major claim to fame and his capital for further research and development at his laboratory in Menlo Park (see the next section) when he invented the phonograph. Menlo Park itself was built with funds from earlier, lesser-known inventions (most in telegraphy, such as the stock ticker).

Opened in 1887, for more than 40 years Thomas Edison's West Orange laboratory had an enormous impact on society and cultures throughout the world. From this lab came the motion picture camera, improved versions of the phonograph, sound movies, and the nickel-iron alkaline electric storage battery.

Run by the US National Park Service, replete with informative displays and staffed by knowledgeable park rangers, the Edison site should be a top international tourist destination. Visits there even include tours of

### Editor's Note

In this issue, we introduce a trial column on the local geography of computing by Mike Geselowitz, the director of the IEEE History Center in New Brunswick, NJ.

Edison's nearby mansion, Glenmont. Although the Edison Site is currently closed for major renovations, when it reopens in late 2006 it will be even more spectacular.

### **Thomas Alva Edison Memorial Tower and Menlo Park Museum**

Just a quick 20-mile drive down US Route 1 from West Orange will bring the traveler to the site of Edison's previous laboratory in Menlo Park (since renamed Edison in his honor). The site features a bulb-shaped tower and a museum that, although more modest than the one maintained by the National Park Service, is nevertheless worth a visit. More than 400 patents were issued to Edison while he worked at this site, including his improved incandescent lamp (the light bulb) and the carbon-button transmitter.

His work of most interest to the historian of computing, however, is considered Edison's only scientific discovery (as opposed to an engineering development)—the Edison Effect. Edison noted in 1883 that when a lamp was fitted with a filament and a plate, an electric current flowed through the vacuum. Although he saw no real practical use, he did apply for a patent on the device as a voltage regulator, and the research was presented at the first meeting of the American Institute of Electrical Engineers (the predecessor to the IEEE) in 1884. The Edison Effect eventually led to the Fleming Valve (diode) and to electronics—the foundation of modern computing.

### **InfoAge Science/History Learning Center**

Heading south from Edison for 35 miles on the Garden State Parkway will bring travelers to the InfoAge Science/History Learning Center in Wall, on the site of the former Camp Evans. Evans, the site of the US Army Signal Corps' major electronics research laboratory, was a key player in the development of radar in World War II, the beginnings of satellite telecommunications (Project Diana was based there), the application of the transistor, and many more technologies. Interestingly, before the army took it over, it was a Marconi station where Edwin H. Armstrong, Ernst F.W. Alexanderson, and David Sarnoff (see the next section) did important radio research. The site is now on the US National Register of Historic Places. As a new museum, it's just gearing up to become a regional center for disseminating the history of technology.

### **David Sarnoff Library**

Fifty miles almost due west of Wall is Princeton. Of course, Princeton University is

## **Museum Information**

For more information on the museums cited in this article, please see the following:

- Historic Speedwell, 333 Speedwell Ave., Morristown, NJ 07960, +1 973 540 0211; <http://speedwell.org/>
- Morris Museum, 6 Normandy Heights Rd., Morristown, NJ 07960, +1 973 971 3700; <http://www.morrismuseum.org>
- Edison National Historic Site, Main St. and Lakeside Ave., West Orange, NJ 07052, +1 973 736 0551; <http://www.nps.gov/edis/index.htm>
- Thomas Alva Edison Memorial Tower and Menlo Park Museum, 37 Christie St., Edison, NJ 08820, +1 732 549 3299; <http://www.menloparkmuseum.com/>
- InfoAge Science/History Learning Center, 2201 Marconi Rd., +1 732 280 3000; <http://www.infoage.org>
- David Sarnoff Library, 201 Washington Rd., Princeton NJ 08543, +1 609 734 2636; <http://www.davidsarnoff.org/>
- New Jersey Museum of Agriculture, 103 College Farm Rd., North Brunswick, NJ 08902, +1 732 249 2077; <http://www.agriculturemuseum.org/info.html>.

home to many wonderful tourist attractions, but the historical visitor may wish to stop outside the campus at the David Sarnoff Library. On the grounds of the Sarnoff Corporation (now part of SRI International, formerly a division of RCA), the library houses memorabilia and exhibits about the life and accomplishments of the radio and television pioneer. The exhibits include the histories of technologies in which he had a hand, such as solid-state circuits and electro-optics.

### **New Jersey Museum of Agriculture**

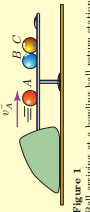
Back on US Route 1, the visitor can go 15 miles north of Princeton to end in North Brunswick, just across the river from the starting point in Piscataway. The North Brunswick-New Brunswick-Piscataway area is home to the sprawling campuses of Rutgers, the State University of New Jersey. Like Princeton, this university is full of interesting attractions. This tour will end with just one—the New Jersey Museum of Agriculture. A museum of agriculture might seem a stretch for the historian of modern electrical and electronic technologies, but this museum happens to curate the world's largest collection relative to rural electrification. Without the spread of power, communication, and computation, all the other technologies we take for granted would not be possible.

Readers may contact Michael Geselowitz at [m.geselowitz@ieee.org](mailto:m.geselowitz@ieee.org).



322 Momentum Methods for Particles Chapter 5

**Example 5.8**



**Figure 1**  
Ball striking at a bowling ball return station.

**Interesting Fact**  
**Propagation of Information.** How does ball  $C$  know about the collision between balls  $A$  and  $B$ ? The answer is: partly because  $B$  (as a whole) will move and partly because the disturbance travels through  $B$  and are then transmitted to  $C$ . Because of this dual information propagation mechanism, if the sound information travels much faster than  $B$  will as a whole, we can claim that  $C$  is affected by the  $A$ - $B$  impact even if  $B$  does not appreciably move. When this happens, we then analyze the combined impact using the conservation of momentum in the frame in which objects can change velocity without changing position. Therefore, in solving this problem we have assumed that speeds of the order of  $6\text{ ft/s}$  (e.g.  $v_A$ ) are much smaller than the speed of sound in the bowling balls, which is estimated to exceed the speed of sound in common nylon products, this speed being roughly  $3000\text{ ft/s}$ .

**item A**

**Solution**  
**Read being.** As stated, this problem lies outside the theorem we have presented in this section (as are all general multibody simultaneous impacts). In fact, our theory was developed for the impact of only two particles at a time. However, the generalization to multibody simultaneous impacts is not straightforward and is somewhat very slightly off and make the problem tractable. So, effectively, if we assume that balls  $B$  and  $C$  are separated by a thin gap, instead of being in contact, then the problem can be viewed as a *collision sequence*. This sequence consists, at the very least, of an  $A$ - $B$  impact followed by a  $B$ - $C$  impact, although, as we will see, additional impacts may occur. The final motion of the balls must satisfy the inequality:

$$(v_A^*)_x \leq (v_B^*)_x \leq (v_C^*)_x, \quad (1)$$

where the subjection is shown in Fig. 2. By demanding that the above inequality be satisfied, we will be certain that all impacts have occurred and that the problem is “done”.

**Modeling** The fact that all the balls have the same diameter means that the LOI is parallel to the guides of the ball return. Therefore, the reaction forces between the balls and the guides are not affected by the impact and can all be considered *non-impulsive* so that the resulting impact-relevant FBDs for the  $A$ - $B$  and  $B$ - $C$  impacts are those in Fig. 2. Furthermore, as the motion is one-dimensional parallel to the LOI, all the impacts are unconstrained direct central impacts.

**item B**

**Governing Equations**

**Newton-Euler/Balance Equations** At the very least, we will have the two momenta,  $A$ - $B$  and  $B$ - $C$ , and the conservation of linear momentum for these collisions as

$$m_A(v_A^*)_x + m_B(v_B^*)_x = m_A(v_A^*)_x + m_B(v_B^*)_x, \quad (2)$$

$$m_B(v_B^*)_x + m_C(v_C^*)_x = m_B(v_B^*)_x + m_C(v_C^*)_x, \quad (3)$$

where  $(v_A^*)_x$  and  $(v_B^*)_x$  are the velocities of  $A$  and  $B$  right after the first collision;  $(v_B^*)_x$  is the velocity of  $C$  right before the second collision; and  $(v_B^*)_x$  and  $(v_C^*)_x$  are the velocities of  $B$  and  $C$  right after the second collision. The reason we are not calling  $(v_A^*)_x$ ,  $(v_B^*)_x$ , and  $(v_C^*)_x$  the final “post-impact” velocities is that, up to this point, we have not yet verified that these velocities satisfy the inequality in Eq. (1).

Figure 1: Poopoo.

**Conservative Systems** *Conservative forces* are forces whose work is independent of the path (i.e., the work only depends on the endpoints of the path) and for which there exists a scalar potential energy. A *conservative system* is one for which all the forces doing work are conservative. For conservative systems, the balance of work and energy becomes the *conservation of mechanical energy*, which is expressed by

**item C**

Eq. (4.33), p. 230

$$T_1 + V_1 = T_2 + V_2.$$

Important conservative forces include elastic spring forces and gravitational forces (constant or not). The *potential energy of a linear elastic spring force* is given by

Eq. (4.35), p. 230

$$V_s = \frac{1}{2}k\delta^2,$$

the *potential energy of a constant gravitational force* is given by

Eq. (4.34), p. 230

$$V_g = mgh,$$

and the *potential energy of the force of gravity* is given by

Eq. (4.36), p. 231

$$V_G = -\frac{Gm_A m_B}{r}.$$

Finally, if we have a system in which there are both conservative and non-conservative forces doing work, we can use Eq. (4.18) or we take advantage of the potential energies of the conservative forces by writing the balance of work and energy as

Eq. (4.17), p. 232

$$T_1 + V_1 + (U_{1 \rightarrow 2})_{nc} = T_2 + V_2,$$

where  $(U_{1 \rightarrow 2})_{nc}$  is the work done by forces for which we do not have a potential function. It is this equation that we will use when there are forces doing work for which we have not derived a potential function.

**item D**

**Concept Alert**  
Elastic potential energy is always positive. Whether a spring is elongated or compressed, i.e., whether  $\delta$  is positive or negative, the potential energy of a spring is always positive!

**Helpful Information**  
When should you use the balance of work and energy? The answer to this question is actually pretty simple if we look at the general form of the balance of work and energy as given by Eq. (4.15). This equation tells us how the speed of a particle changes as a force  $F$  causes it to move between 0 and 2. Therefore, we can use it in situations where we are interested in the speed of a particle and changes in speed of a particle.

# L<sup>A</sup>T<sub>E</sub>X at a liberal arts college

Jon Breitenbucher  
Department of Mathematics and Computer Science  
308 E. University  
The College of Wooster  
Wooster, Ohio 44691 USA  
e-mail: [jbreitenbuch@wooster.edu](mailto:jbreitenbuch@wooster.edu)  
web: <http://jbreitenbuch.wooster.edu/~jonb>

## Abstract

Does L<sup>A</sup>T<sub>E</sub>X have a place in a liberal arts education? Yes, and in this article I present my reasons for introducing L<sup>A</sup>T<sub>E</sub>X in an undergraduate liberal arts setting. I also present how I introduced L<sup>A</sup>T<sub>E</sub>X, issues that were encountered, and what students and faculty think the impact has been.

## 1 Background

The College of Wooster is a small liberal arts college located in northeastern Ohio. Wooster's annual enrollment is around 1800 students and the Mathematics and Computer Science Department typically has 25–30 majors per year. One of the distinguishing features about Wooster is its Independent Study (IS) program founded by its seventh president Howard Lowry. The independent study program requires every senior to complete a year long independent research project. A typical faculty member will advise 3–4 such projects. One of the challenges I encountered in my first advising experience was getting students to write technical mathematics. This is what lead me to introduce L<sup>A</sup>T<sub>E</sub>X at Wooster.

The majority of seniors use Microsoft Word. This probably comes as no surprise to most of the readers. In fact, for the vast majority of seniors, Word is the appropriate tool. However, Word is not the best tool for everyone. Students writing in foreign languages that requires special fonts and the ability to have text go from right to left could benefit from using  $\LaTeX$  in conjunction with  $XeTeX$ . Students in music might find  $MusiXTeX$  to be a better environment for preparing their theses. However, I think the science majors have the most to be gained from switching to  $\LaTeX$ . Science majors typically have a large number of equations, figures, and tables. Having used  $\LaTeX$  for my dissertation, I knew that it could do a much better job of formatting the theses of the science majors. I decided to introduce  $\LaTeX$  into my department first and then to expand into other departments. In this article I would like to outline my approach, some of the issues that I encountered, and student reaction.

## 2 Why $\LaTeX$ ?

As a student at Wooster I had struggled with Word version 5.5 on the Mac to produce a passable document. Some of you may remember that the earlier versions of Word were not too different from  $TeX$ . One would type in a command sequence to get a sum, product, or other symbols. It may even be the case that some of these command sequences survived into the present day, but people have long forgotten their existence.

When I returned to teach at Wooster I was shocked by the poor quality of the Independent Study theses. Current students were not able to produce a document that looked anywhere near the quality of my thesis from nine years earlier, and this was with more advanced versions of Word. I found that the students spent weeks trying to format their theses to make sure that section numbering, equation numbering, and figure numbering were correct (and most did not succeed). Almost none of the students knew how to have an automatic Table of Contents, Figures, etc. created by Word. It was at this point that I asked myself whether  $\LaTeX$  could make the process of writing a thesis more about the writing and less about the formatting. Why  $\LaTeX$ ? Because I felt like it was strong in all of the areas with which students were having trouble.  $\LaTeX$  would handle the numbering, formatting, front matter and back matter and the students could just worry about the content.

However, there is a down side which Neuwirth touches on in [Neuwirth \[1991\]](#); none of the students know  $\LaTeX$ . This means that someone has to be willing to teach them and answer their questions. However, this situation is different than that addressed by Neuwirth. Neuwirth was discussing the place of  $\TeX$  in what would be considered middle school and high school in the United States. And I agree that those students don't need the full power of  $\TeX$ , but I'm not sure that they couldn't benefit from an introduction to  $\LaTeX$ . One of the questions that my experiences have raised is, "Where do our students learn how to use Word or other document preparation software?" I have been unable to find anyone that knows the answer to this question. Some of my colleagues assumed that our Writing Center was helping students learn how to write technical documents, but in talking to the Writing Center staff I found this was not the case. What we have found is that most of our math and science students begin college or university study with no idea of how to use Word or other tools to write a technical paper. However, they are expected to be able to produce a technical paper when they get to graduate school. With this being the case, then it is incumbent on us to teach them, and so that is what I decided to do with the students at Wooster. There is no release time or other compensation for this; I do it because I love doing it.

### 3 The Process

So how did I go about getting  $\LaTeX$  into our program? The first step was to identify exactly what I wanted to accomplish. As mentioned above, I really want the students to let  $\LaTeX$  handle all of the formatting. What does that mean? I decided it means that I don't want students to have to load packages, learn the intricacies of incorporating graphics, or have to try to force  $\LaTeX$  to do something that Word can do. What this really means is that I needed to construct a Wooster thesis class. (I leave the explanation of the difference between a class and package to a more knowledgeable  $\TeX$ pert.)

Before trying to construct a thesis class for Wooster, I examined a number of classes available at other institutions. During this process I discovered two things: none of the classes did exactly what I wanted and almost all of them were modifications of the standard book or article class. After realizing this, I decided to

try to modify the book class myself using a couple of other thesis classes<sup>1</sup> as models. To meet the stated goal above, my class has to load all of the packages I think students will need or provide a class option which will load certain packages. At first I was only loading a few packages, but as students have used the class I have added more packages and options. I think the current mix<sup>2</sup> serves my students very well, and I don't envision them needing any more packages. This process was not easy and I wish I knew and had known more about writing a class file. I would recommend that a beginner or intermediate user find a T<sub>E</sub>Xpert to help them write or modify a class file. Doing so will save a lot of hair pulling and time spent in trial and error.

At this point I sought input from my colleagues, the Registrar, the Secretary of the College, and the Vice President for College Relations and Marketing to make sure that the format and images used were acceptable. Others might not need to include so many people, but since IS is such a major component of our curriculum, I needed to make sure everyone liked the design. I was told to change a few things and resubmit, at which point my design was approved. Others will probably find that they will have a similar experience. Now it was time to involve the students.

### 3.1 Editors, platforms, and documentation, oh my!

There are a few things that I had to do before I could start showing students how to use L<sup>A</sup>T<sub>E</sub>X and my class file. The first is dealing with the platform issue. I am very committed to allowing users to choose the operating system they are comfortable with using. I have almost 20 years of experience on the Mac OS and as such I know almost all the T<sub>E</sub>X editors available. On the other hand, I don't know much about Linux and have only seven or so years of experience with Windows. So my first task was to identify software packages for each of the three major OS variants. If you find that you need to do this keep in mind that the school will probably not want to buy software, so you need to find free or low-cost shareware solutions. After some research I settled on the following: TeXShop/GW<sub>T</sub>E<sub>X</sub> for OS X, TeXnicCenter/Mik<sub>T</sub>E<sub>X</sub> for Windows XP, and Kile/te<sub>T</sub>E<sub>X</sub> for GNU/Linux. Why these packages? I chose these packages because they all provide panels or menus

---

<sup>1</sup>I used the kthesis and osuthesis classes as models.

<sup>2</sup>My class loads ifpdf, amsthm, amssymb, amsmath, setspace, graphicx, eso-pic, natbib, float, caption, subfig, hyperref, and color and has options for pxfonts, floatflt, and listings.

for common  $\text{\LaTeX}$  tasks, are free, and are as close to the point-and-click Word model as I could find. They also do not require nearly as much technical ability to install and use as something like Emacs. Remember a number of the students may not be technically savvy, so the more like Word the better. Emacs is great and would have made for a more uniform environment, but I was afraid the level of technical ability required to install and use Emacs would scare away the weaker students (the ones I most want to use  $\text{\LaTeX}$ ).

Once I settled on software packages I am comfortable supporting, it was time to document the thesis class and introduce  $\text{\LaTeX}$ . I chose to document  $\text{\LaTeX}$  and the thesis class by using the thesis class to write the documentation. In this way I am able to give students a zip archive containing all the files needed to produce the documentation. In addition the students can use the archive as a template for creating their theses; they just need to make a copy of the folder they get when they unzip the download and start putting their content into the files. This has worked very well as they can see the code that I used to achieve something and copy and paste or alter it to their needs.

My documentation<sup>3</sup> covers very basic things like starting new chapters and sections, creating lists, making things bold or italics, including graphics, inputting mathematics, and inputting computer code and does not cover installation of a  $\text{\TeX}$  system or the software (that is left to the authors of the software). It is really a summary of things found in [Kopka and Daly \[2003\]](#), [Mittelbach et al. \[2004\]](#) and [Flynn \[2003\]](#), except for the Typesetting Mathematical Formulae chapter which comes from [Oetiker et al. \[2003\]](#). The intent is for students to teach themselves how to use the few  $\text{\LaTeX}$  commands that they need and to come to me if they have difficulty. Choosing editors with panels or menus for  $\text{\LaTeX}$  input makes this possible. This is a much different approach than used by [Gray and Costanza \[2003\]](#) and [Childs \[1989\]](#) where there is an actual course where students learn  $\text{\TeX}$  or  $\text{\LaTeX}$ . Students have done reasonably well under my setup, but a course where some introductory  $\text{\LaTeX}$  could be covered would be desirable. My department is considering trying to move technical writing issues into the proofs/introduction to higher level mathematics course, but it is hard to cut content in favor of this new material.

---

<sup>3</sup>[http://jbreitenbuch.wooster.edu/pdf/latex/IS\\_guide.pdf](http://jbreitenbuch.wooster.edu/pdf/latex/IS_guide.pdf)

## 3.2 Involving others

In writing the class file I tried to make it as general as possible to allow other departments to use it. After a year of use in the Math and CS department, I introduced the class file and L<sup>A</sup>T<sub>E</sub>X to the physics students. The students picked up L<sup>A</sup>T<sub>E</sub>X very easily and liked the results. They particularly liked the fact that I had set everything up to use pdfT<sub>E</sub>X and produce a “live” document. However, some of the physics faculty did not like the design of the output and so they have modified the class to produce a different-looking document. Others trying to introduce L<sup>A</sup>T<sub>E</sub>X may find this as well. Make sure everyone knows you are not responsible for modifying the class file or troubleshooting others’ changes; otherwise you will find yourself maintaining ten slightly different versions of the same class.

After introducing L<sup>A</sup>T<sub>E</sub>X to the physics students, I approached Chemistry and Biology. My plan was to move through the sciences and then approach Music, Classical Studies, Chinese, and Arabic. I discovered that neither the Chemistry nor the Biology department were interested in introducing this to their students, their main concern being that no one in their departments was familiar with L<sup>A</sup>T<sub>E</sub>X. I met with the same response from Classical Studies, but in this case no one had even heard of L<sup>A</sup>T<sub>E</sub>X. This is a real issue when trying to introduce L<sup>A</sup>T<sub>E</sub>X. In retrospect, I should have identified a few individuals in each department to learn L<sup>A</sup>T<sub>E</sub>X from me. Those people would then act as point people for their students and would use me as backup. A faculty workshop designed around the material of [Gray and Costanza \[2003\]](#) might be a way to accomplish this.

So, as it stands now, Math and CS and Physics are the only departments using L<sup>A</sup>T<sub>E</sub>X, which is not surprising when one browses through the various mailing lists and samples peoples’ fields-of-study. Involving people from other departments from the start might have made a difference. I would suggest that if others try this that they develop a clear plan for implementation and have a timeline to measure progress.

## 4 Assessment

So how did I do? That’s hard to say because I didn’t have a formal assessment plan in place. My assessment has been in the form of an informal Pizza Party after all the seniors have completed their theses, and two questions on the departmental IS

evaluation form. This is not what I would recommend for others. Unfortunately, I am beyond the stage for assessing the success of the introduction, and have lost that chance. What I am doing is developing materials to assess the process of learning  $\text{\LaTeX}$  so that I can improve that process and make it easier for students.

There are a few things that I can communicate in an anecdotal manner. In general the students have felt that this model is working well. The first group of students suggested introducing  $\text{\LaTeX}$  earlier in the curriculum. I took that recommendation and created a homework package and template file and encourage sophomores to use it and require juniors to do one assignment in  $\text{\LaTeX}$ . I have also started requiring all homework submissions to be typed in sophomore-level classes and above. Students also suggested the need for various capabilities for placing images and styling chapter headings. I incorporated the packages necessary to accomplish this in the class file. The result has been that no one had any suggestions at this year's pizza party.

The students also felt like they did focus more on the writing, but there are some formatting issues that really bother them. Image placement is a big source of frustration. The students are used to dragging an image into the document exactly where they want it and having it stay. It takes them some time to get used to letting the images float and to use references to refer the their images. The other frustration is learning commands. It takes them a few weeks to really get the hang of things. However all of them said these minor issues are more than compensated for by the auto-generation features of  $\text{\LaTeX}$ , and they are glad they took the time to learn  $\text{\LaTeX}$ .

Has this process improved the writing? This is difficult to answer. The questions I used to measure this on the IS evaluation are:

- Based on your discussions with this IS student, the bibliography, and the final written document, which statement best describes the student's assimilation of the material?
  1. The student assimilated material from a wide variety of sources.
  2. The student used material from multiple sources and did some assimilation of that material.
  3. The student used material from multiple sources.
  4. The student primarily used material from one source, but did use some material from at least one other source.



5. The student used one primary source from which all material is taken.
- Based on the final written document, which statement best describes this IS?
    1. The IS is written in a clear and well-organized manner, with excellent grammar, spelling, and typesetting. Moreover, it is written in the student's unique style and directed toward an audience of peers.
    2. The IS is very readable, with very few errors in spelling, grammar, or typesetting. The thesis is well-organized.
    3. The IS is readable, despite some errors in spelling, grammar, or typesetting. The thesis is well-organized.
    4. A number of errors in spelling, grammar, or typesetting make this IS somewhat difficult to read. A better organization of ideas would have made it more clear.
    5. The IS lacks organization, the grammar is poor, and it is difficult to read.

I chose these questions because my goal is to make the IS experience more about the writing and less about the formatting. If I am succeeding then students using L<sup>A</sup>T<sub>E</sub>X should assimilate more and produce a better written document. Of course I cannot set up a control group and conduct a true study to control for all the confounding factors, but anecdotally I can say that, in general, students using L<sup>A</sup>T<sub>E</sub>X have scored better on these questions than those who have not. My colleagues also agree that in their judgement L<sup>A</sup>T<sub>E</sub>X has increased the overall quality of the IS produced by the students.

So I would say that my attempt has accomplished my goal. For anyone planning on doing something similar, an assessment plan for all phases is a must. I say this because more and more accreditation bodies want to see evidence showing the success or failure to meet stated goals. Also, I do not think that you have to have a senior thesis to try this. Programs with writing across the curriculum could also see an improvement in student performance, and might have an easier time of assessing L<sup>A</sup>T<sub>E</sub>X's impact.

## 5 The Future

So now what do I do? There are a few things I hope to do in the next few years. One is to expand the use of  $\LaTeX$  into the foreign languages. The introduction of  $XeTeX$  and OS X makes it extremely easy to typeset in foreign languages. I think that students studying Eastern languages would find great benefits to using the  $XeTeX$  system, and I hope to be able to talk to the faculty in those disciplines in the near future. Another goal is to increase the use of  $\LaTeX$  in lower level courses, which will require training my colleagues in the use of  $\LaTeX$  and will allow students to learn  $\LaTeX$  at a much slower pace.

## 6 Acknowledgments

I want to thank Karl Berry for encouraging me to write about my experience and providing several articles relating to the topic. I thank the reviewers for their insightful comments. And most of all I thank all of the Wooster students who have used  $\LaTeX$  and my class file for their ISs and provided valuable feedback, without them this project wouldn't exist.

## References

- Bart Childs. Teaching  $\TeX$ . *TUGboat*, 10(2):156–163, 1989.
- Peter Flynn. Formatting Information. *TUGboat*, 23(2):115–237, 2003. URL <http://www.tug.org/tex-archive/info/beginlatex/beginlatex.letter.pdf>.
- Gary Gray and Francesco Costanza. Experiences and lessons learned teaching  $\LaTeX$  to university students. *TUGboat*, 24(1):124–131, 2003.
- Helmut Kopka and Patrick W. Daly. *Guide to  $\LaTeX$* . Pearson Education, New York, 4th edition, 2003.
- Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The  $\LaTeX$  Companion*. Addison Wesley Professional, New York, 2nd edition, 2004.

Konrad Neuwirth.  $\TeX$  in Schools: Just Say No. *TUGboat*, 12(1):171–173, 1991.

Tobias Oetiker, Hurbert Partl, Irene Hyna, and Elisabeth Schlegl. The Not So Short Introduction to  $\LaTeX$ 2 $\epsilon$ . 2003. URL <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>.

# T<sub>E</sub>X and medicine

Elizabeth Dearborn

55 Lark Street

Buffalo, New York 14211

U.S.A.

[elizabeth@medicalese.org](mailto:elizabeth@medicalese.org)

<http://www.MeDiCaLeSe.org>

## Abstract

The elegance and precision of T<sub>E</sub>X make it a good choice for printed materials in the field of medicine. This article describes my adventures in teaching myself enough about T<sub>E</sub>X to successfully produce a non-standard medical reference book.

## 1 Introduction

Way back in 1985, when I worked at a struggling graphic design shop, I learned to set type on a Mergenthaler phototypesetting machine which was old even then. Remember width cards and photographic chemicals? In 1987, I went into medical transcription, which is an extremely knowledge-intensive line of work with very tight deadlines. I wanted to be good at what I did, so I maintained my own word list, which grew exponentially with time.

I was also interested in computer programming and the internet. I studied these things on my own, and in 2001 I started my first website at <http://www.MeDiCaLeSe.org>. Using a javascript site search program I found on the internet, I put my word list online in searchable form. Although the website and the database were available to everyone, my main objective at the time was to be able to search my own word list quickly. Whenever a person searched the database, my entire word list was loaded into a temporary file on the client's machine. This also required javascript to be enabled on the client side.

After a couple of years, the word list had gotten huge, the javascript search engine was taking a long time to load, and I was beginning to feel that the products of my research were worth money. I wanted to find a way to serve *only* the requested results and serve them up faster, and, if possible, I wanted to accomplish this without making my code visible to the end user.

## 2 The php programming language

I looked, but I could not find a free or inexpensive package that would do what I wanted. Hiring a programmer was out of the question, as was going back to school. I'm not the type who takes courses; everything I've ever wanted to know about, I've learned on my own, with some degree of success. I checked

out the various programming languages which are used on the internet, such as C++, Pascal, php, and Perl. Of these, php seemed best for my purposes. It is free, runs entirely on the server side, and virtually every web host offers it to its customers.

My approach to learning enough php to get the job done was very similar to the way I later went about learning T<sub>E</sub>X. First, I defined exactly what I wanted to do. Next, I searched the official php website at <http://www.php.net> to see if a command existed that would accomplish this. If not, I would need to write my own. Sound familiar?

I purchased two books both published by O'Reilly Media, *Programming PHP* by Rasmus Lerdorf and Kevin Tatroe, and *PHP Cookbook* by David Sklar and Adam Trachtenberg. Between these two wonderful books, the vast resources of the internet, and my own determination, I had my improved site search ready to go live in about two months' time, in the summer of 2003. The original version of my php site search had the Google-style highlighting as it does today, but did not then include the capability to exclude one string from the search results. I invite you to visit <http://www.MeDiCaLeSe.org> and try it out.

## 3 A database in plain text files

Type the word "Jones" into the search box. This will return 12 results, two of which are shown below:

Matches: jones: 12

**Orthopedics:** Jones fracture - fractured proximal fifth metatarsal.

**Diagnostics:** Jones silver stain.

If you view the page source, you won't see a listing of the php code that made the page, but you

will be able to see the HTML code for the results. This is not very different from what the plain text entries in the database actually look like:

```
<B>Orthopedics: </B>Jones fracture -
fractured proximal fifth metatarsal.|
06/04/04
```

— and so on until:

```
<B>Surgery: </B>Smead-Jones continuous
mass closure.|06/02
```

Short and sweet, and in no particular order. Anything following the vertical line does not show up on the results page.

I do extensive research on medical terminology every single day, and I enter the new or changed data into the two text files which make up the database and upload them to the server every night. The php search program calculates the number of words, number of entries, and the last time the files were updated.

Why did I make plain text files? Because I was afraid of MySQL. I honestly don't know whether this hurt or helped when it came time to publish the database in book form!

#### 4 Show me the money!

I worked hard for little compensation as a transcriptionist, and in the case of the website for no compensation at all. In 2004 someone suggested I turn my word database into a book. At first I resisted, because the daily updates and corrections were a large part of the value of the website, and I thought it was impossible to publish even a reasonably current book on such an enormous, constantly changing, and complicated subject, if typographic quality were to be a consideration.

I had seen a few quickly-printed books which were published in a hurry to cash in on current events, but their quality was atrocious — full of typographical errors and crookedly printed on brittle, yellowing newsprint.

I had guest-edited several medical terminology books for a large medical publisher in the mid-1990s. These books were very nicely produced, but as a transcriptionist I felt the information they offered was not quite enough. And, by now, I enjoyed having total control over my own website and was not about to relinquish that just to get a book published. If I went to the trouble to put a manuscript together, I felt that the big medical publishers would turn it

away. I was good at internet research, and I started looking into print-on-demand publishing.

I would have to typeset my own manuscript, but this didn't intimidate me, since I had worked in typesetting before. I had heard of T<sub>E</sub>X in the context of mathematical typesetting, and I started studying it on the internet. I didn't have much money, and I appreciated the fact that T<sub>E</sub>X, besides being free, requires fairly minimal computer hardware. I continued to study T<sub>E</sub>X while organizing my database in manuscript form. I downloaded and experimented with T<sub>E</sub>X, joined T<sub>E</sub>X Users Group, and lurked on `comp.text.tex` and the T<sub>E</sub>Xhax list.

I write a little bit of short mystery fiction, and I knew several mystery writers who had either published with a subsidy house or gone out on a limb and published their own work. Subsidy publishing didn't interest me, as I needed to have the books available through the regular book-buying channels. Finally, a friend who started her own publishing company convinced me that I could do the same, and that I could have the books printed on demand by Lightning Source and distributed worldwide by Ingram. This was all I needed to hear! In April 2005 I obtained a business license and business checking account in the name of Blowtorch Press, and ordered a block of ISBN numbers. I named the book *MeDiCaLeSe 2005*, so that identifying subsequent revisions would be a no-brainer.

#### 5 Why T<sub>E</sub>X for medicine?

The Computer Modern font has almost all the diacritical marks and special characters that are needed in medicine. The `textcomp` package is needed only for the `textrecipe` (Rx) symbol. I wanted the  $\mu$  symbol, the dot and umlaut in Åström, and the beta symbol in Dia $\beta$ eta to be properly typeset, even though these characters are not used in medical transcription; in that context, these words are written as micro, Astrom, and DiaBeta respectively. Medical transcription is usually done in a word processing program, and turnaround time is crucial.

By now I had found the T<sub>E</sub>X editor I like best, which is L<sup>A</sup>T<sub>E</sub>X Editor by Shu Shen, a graduate student in Singapore. This software is free and very lightweight, which was an advantage since I didn't know much about T<sub>E</sub>X beyond what was needed to get the job done.

The php code does all the heavy lifting for the website. For a book, I would need to organize the data somehow. I put the entries into 39 chapters called Drugs, Abbreviations, Vocabulary, Cardiology, Neurology, and so on, alphabetizing the entries within the chapters, and then duplicated the entries

into the different chapters as needed. I aimed for as much redundancy as possible, because my primary intended audience of medical transcriptionists would not be willing to purchase the book unless they knew they could find what they were looking for quickly, and with enough information to know whether the word was the correct one for the situation.

Also because of the special requirements of the book's intended audience, I included this statement in the preface:

NO ADDED HYPHENS: We did not introduce new breaks into any of the words in this book. If you see a hyphenated word at the end of a line, it means the word should always be written with the hyphen. For the medical transcription community, we felt it was important to be clear on this, even at the expense of aesthetically undesirable line breaks. We have tried to make the book attractive and easily readable in spite of our self-imposed constraints on hyphenation.

I immediately saw that I would have no end of trouble unless I came up with a bulletproof method of producing dictionary-style pages in double columns. I also needed the first and last words to appear at the top of the page on which they were defined. I began an intensive study of the `fancyhdr`, `geometry`, and `multicol` packages, and I tweaked the code until I was able to produce dictionary-style pages.

Some of the main L<sup>A</sup>T<sub>E</sub>X file for the book appears below. I included my personal commands, which all begin with “en” plus one letter.

```
% c:\m05\medicalese.tex
\documentclass{book}
\def\enl{\filbreak\small\textbf}
\def\ene{\filbreak\normalsize\textbf}
\def\enn{\normalsize}
\def\eno{\enspace\scriptsize}
\def\enc{\filbreak\small\texttt}
\def\enb{\protect\raisebox{-2pt}}
\def\enu{\protect\raisebox{2pt}}
\usepackage{fancyhdr}
\usepackage[papersize={6.14in,9.21in},
margin={0.5in},top={1in},bottom=
{0.75in},headheight={34.545pt},
centering,verbose=true]{geometry}
```

```
\usepackage[none]{hyphenat}
\usepackage{textcomp}
\usepackage{multicol}
\newcommand{\enk}[1]{#1\markboth
{#1}{#1}}
% subscript: \small CO\enb{2}
% greek letters: $\mu$ produces the micro
% symbol
% umlaut: Waldenstr"\{o}m's
\begin{document}
% \frontmatter
\mainmatter
% COPY THIS CODE WHEN CHANGING HEADERS
% \end{multicols}
% \eject
\input drugs
\input notes
\input abbrev
\input notes
\input vocab
\input notes
\input institut
\input notes
\input studies
\input notes
\input drugclass
\input notes
\input vaccines
\input notes
\input anatomy
\input notes
\input diseases
\input notes
...
\input wounds
\input notes
\input discont
\input notes
% COPY THIS CODE WHEN CHANGING HEADERS
% \backmatter
\input notes
\end{document}
\bye
```

The beginning of one of the chapter files:

```
% c:\m05\labpath.tex
\pagestyle{fancy}\fancyhead{}
\fancyhead[LE,R0]{\textsf{\rightmark
\\leftmark}}
\fancyhead[C]{\textbf{Diagnostics}}
\fancyhead[RE,L0]{\textsl{\small includes
laboratory,\\pathology, and radiology}}
\renewcommand{\headrulewidth}{0.2pt}
\fancyfoot[LE,R0]{\textit{MeDiCaLeSe-
```

```

2005}}
\fancyfoot[RE,LO]{\textsf
{www.medicalese.org}}
\renewcommand{\footrulewidth}{0.2pt}
\raggedright
\begin{multicols}{2}\
% END OF CODE TO COPY WHEN CHANGING
% HEADERS ... diagnostics
\textbf{\enk{3-androstenedione}}\enn:
endocrine test used in workup for women
with hirsutism.\ene
{\enk{3D color Doppler}}\enn: developed at
University of Heidelberg, to visualize the
most severe mitral regurgitations, which
produce eccentric jet flows so complex
that they cannot be measured by standard
2D imaging.\ene

```

The end of the same file:

```

{\enk{Xplorer}}\enn~filmless radiographic
imaging system.\ene
{\enk{ZstatFlu}}\enn~throat swab, a quick
test for influenza A and B.\ene
\\
% COPY THIS CODE WHEN CHANGING HEADERS
\end{multicols}
\eject

```

which produces:

**Xplorer** filmless radiographic imaging system.  
**ZStatFlu** throat swab, a quick test for influenza A and B.

## 6 Ugliness and badness

Because of the rule about not adding hyphens, the book contains some ugly paragraphs, like this:

**Avalide:** irbesartan/  
hydrochlorothiazide, combination  
angiotensin II receptor blocker/thiazide  
diuretic.

Underfull \vbox (badness 10000) has occurred while \output is active happened quite often. I learned to disregard it. Here's one that doesn't look that bad, but TeX complains that the line is just a tad too long:

**Markham-Meyerding** hemilaminectomy retractor.

```

Overfull \hbox (4.52963pt too wide) in
paragraph at lines 689--689
\OT1/cmr/bx/n/10 Markham-Meyerding
\OT1/cmr/m/n/10 hemil-aminec-tomy

```

In a project such as this, the most insidious kind of badness is never seen until the .pdf file has been made and one is examining it page by page, since there is no way to know in advance where the page breaks will occur. I'm embarrassed to admit that I didn't catch this point size error in the guide words at the top of page 592 in time.

SaphLITE  
Songer

When something like this happens, go back to the input file and find the entry corresponding to the *first* guide word on the offending page. Just before the two closing brackets that define the first guide word, add the command to revert to normal size type.

## 7 Finished!

I spent a couple of hours a day for about six months typesetting the book. I gave up my transcription job in August to devote more time to the book, and finished the typesetting on September 14, 2005. The book contained essentially the same information as the website of four days prior. Now I had to tackle the cover! At this point I started studying the **pstricks** package in earnest, and I purchased *The L<sup>A</sup>T<sub>E</sub>X Companion*, which is still the only TeX-related book I've ever bought. Because of memory limitations, I wasn't able to use TeX to make the front cover, but I did use it for the publishing company logo and for the text overlays on the spine and the back cover.

Altogether, the cover took me three weeks to make using Paint Shop Pro 7.04. This is not the printer's recommended software, but it was all I had. Because of memory limitations, I made it in three pieces and pasted it together as a giant 300-dpi .tif file. Lightning Source had provided the bar code file and I pasted this on the back cover. Then I burned the cover .tif file and the .pdf files which made up the interior to CD and mailed it off to Lightning. In about ten days, I received my proof copy by overnight delivery. I knew the .pdf files of the text wouldn't cause problems, but I wasn't at all sure how the cover would turn out. To my delight, it was absolutely beautiful.

# A Wayward Wayfarer's Way to $\TeX$

Stephen Moye

Address American Mathematical Society, Publications Technical Group, 201 Charles Street, Providence, RI 02940, [sgm@ams.org](mailto:sgm@ams.org)  
Organist and Choir Director, Bethany Lutheran Church, 116 Rolfe Street, Cranston, RI 02910

Abstract The amusing recollections of one particular humanities  $\TeX$  user's adventures in  $\TeX$ Land.

## 1 Introduction

I flatter myself that my introduction to  $\TeX$  is a bit unusual and that it may be just entertaining enough to share with you. Let me say at the outset that my training is as a musician: specifically as an organist (rather like another very distinguished member of the  $\TeX$  community...), choir director and sometime composer, and some would say not nearly sometime enough.

I received a Bachelor of Music Degree in organ from Heidelberg College in Tiffin, Ohio, and am a whisker away from a Master's degree in ethnomusicology from Brown University in Providence, Rhode Island. Everything was going along smoothly until one day in 1986 when I found myself in the electronics section of a rather fancy department store. Little did I know what an interesting turn life would take...

## 2 The New Love of My Life

I rounded a corner and came up against a rather plain display featuring a Mac Plus. Love at first sight. It had something fascinating about it, and for some reason seemed completely un-computerlike. It took only a second to get accustomed to the mouse, and in minutes I was drawing (admittedly crude) pictures using MacPaint. MacWrite was a revelation: I could type whatever I liked with no fear of errors as they could be corrected with a simple backspace! Cool! No more whiteout, no more punishing re-typing. Way cool!



What this comes down to is this: I am a dyed-in-the-wool Mac user: I'm not happy if I'm not surrounded by mice, gui's with dialogue boxes, windows, drop-down menus, and sleek styling of the computer itself. The Microsoft Windows environment, and the even more arcane, mysterious and mantra-ridden unix environment are alien to me — or at least they were until a few years ago.

So, a few months after I saw it in the store, I had my own Mac Plus. A few months after that I had an early copy of Aldus Corporation's PageMaker. A few months later still, I had an opportunity to see the documents I was designing in PageMaker printed on an Apple LaserWriter. I simply cannot convey to you the wonder of seeing that first output (it would probably embarrass me now) and the sense of empowerment that my humble little Mac Plus gave me. Not too long after that I made the acquaintance of PostScript and the extraordinary *Colophon 3 Alphabet*, created by Adobe in which PostScript was made to do some wonderful things. As the original readme file put it, "Our intent in distributing this [material] is to inspire and inform" and that it did. All you needed was a \$7,000 Apple LaserWriter to see the results!

In 1988 Quark released XPress<sup>1</sup> in direct competition to PageMaker. I was interested, so I read the reviews and ran across one by Charles Seiter for *MacUser* magazine. At the very end of the review, he mentioned almost in passing with a kind of teasing, throw-away manner that, good as the new page layout programs were, they were nothing compared to the "grand-daddy"<sup>2</sup> of them all, *T<sub>E</sub>X* as exemplified by the program *Textures*.

### 3 A Revelation

Why hadn't I heard of this *T<sub>E</sub>X*, or *Textures* for that matter? I had to have it! Little, of course, did I know what interesting times awaited me.

In no time at all (although the details escape me, and might have been interesting in this time before the internet became the all-knowing and all-powerful thing it is today) I found that *Textures* was sold by Addison-Wesley at that time.

---

1. Founded in 1981, Quark had in its early days developed word processor software for the Apple II and Apple III computers

2. Seiter's word. I tried to find the original article in the *MacUser* archives but was unsuccessful.

## 4 And Then I Wrote a Book

Along the way, I discovered a program called Fontographer, one of the first commercially available font editors. Yet again I was overcome by a feeling of empowerment: I could do things easily and quickly with Fontographer that required tremendous resources and time in traditional type design. I digitized several Goudy typefaces that were then unavailable, and edited others to my liking.

I started to keep a notebook, a kind of *vade mecum*, in which I would jot down notes about things I had discovered, ways of working, tips and techniques that I did not want to forget. Eventually I kept this information in a MacWrite document. One day I was assailed by an attack of hubris and wondered if others might be able to use this information...maybe I could write a book...

So, cutting this down to the bare essentials, I worked on the first few chapters and showed them to Earl Allen at Altsys, the company that at that time developed and marketed Fontographer. He was very enthusiastic and encouraged me to finish it and send it to a publisher. I wrote the rest of the book, and sent a sample chapter to a company that was suggested to me: MIS:Press, then a division of Holt.

A few days later I received a call from the publisher, Paul Farrell. We exchanged pleasantries and established for the moment that we liked each other, for the nonce anyway. Then the conversation took a serious turn.

He said, “Well, most people on the west coast use PageMaker and most people on the east coast use XPress, so what do you use?”

“Uh... er...” I began articulately, “well sir, I use T<sub>E</sub>X.”

There was a moment of silence so profound on the other end of the line that I thought the connection had been broken. I was just about to say something...

“Oh my God,” he said, “bitmaps!”

It was with some difficulty that I convinced him that T<sub>E</sub>X had evolved far beyond the use of bitmap fonts, in fact he’d seen the results with his own eyes in the sample I sent him. The book, *Fontographer: Type by Design*, finally appeared in 1995. Sadly, two years later, IDG International bought MIS:Press. IDG promptly destroyed all copies of all MIS:Press books that it deemed would not sell in numbers on a par with *The Joy of Cooking*. Maybe they did me a favor of sorts: I have seen copies of the book sell for as much as \$300, and an asking price over \$1,000.

So, my world was happy and stable: I authored a book, typeset concert programs, church bulletins and all kinds of documents for non-profit arts organizations. T<sub>E</sub>X was

not about to go away, and Textures had become a valued tool that would be around forever. Wouldn't it?

## 5 Then Along Came Mac OS X

Apple has an annoying penchant for shaking up both itself and its loyal customers. OS X is a good example of this. It wasn't just OS 9 in a new outfit, it was radically different: So different that OS 9 software (such as Textures, for example) had to be used in an emulation of OS 9 on OS X. Blue Sky was quick to point out that Textures worked just fine under "Classic" (the name for the emulation of OS 9 in OS X) and in fact that was true. What became progressively more irritating as time went on was having to open up Classic at all. Every other piece of software I used had been rewritten for OS X. Equally irritating were Blue Sky's reassurances given over a period of *years* that an OS X version was in the works.

And there were problems. It became increasingly difficult to print *reliably* from Textures. As well, the world was moving to Unicode and OpenType, and Textures, operation under OS-9 was stuck using standard PostScript fonts. It is a measure of how good and useful Textures was that we stuck with it for as long as we did.

## 6 "Hey, kid, have I got a job for you..."

The next jog in the road of my journey in T<sub>E</sub>Xland came from a most unexpected source. A choir member at the church for which I was at the time organist and choir director approached me. I knew that she worked at the American Mathematical Society (aka T<sub>E</sub>X-heaven), Victoria Ancona, Manager of Electronic Pre-Press at the AMS. She said to me, in essence, "Boy, have I got a job for you... Do you want to work at the AMS?"

I was dumbfounded and at a loss for words – a rare occurrence for me. Here I was being offered the possibility of working in the same environment as two stellar figures in the T<sub>E</sub>X world: Mike Downes and Barbara Beeton. Actually, as I learned later, I would be working *in the very same department*, the Publications Technical group. I count it a great loss that I never knew Mike Downes very well: Sadly he died before I was able to work up the courage to engage him in conversation – my loss. Getting to know Barbara Beeton has been a treat, even though I know I try her patience from time to time. There is so much to learn!

## 7 As It Was in the Beginning...

I started to see mentions of Dick Koch's new  $\TeX$  editor and previewer called  $\TeX$ Shop, and it sounded interesting. At the time I was still using Textures and my various projects that I had going did not really allow me to change my  $\TeX$  environment. But it sounded promising and I kept an eye out for future developments,

They were not long in coming. Word was circulating about Gerben Wierda's i-Installer, and how it took care of all the painful details of installing and maintaining  $\TeX$ . Better and better. I downloaded both  $\TeX$ Shop and i-Installer and put them to work. Lo! and behold, it worked. This was beginning to look a lot like what Textures was to Mac OS 9. But the best was yet to come...

### 7.1 St. Jonathan and the Blessed $X_{\text{f}}\TeX$

One day I read something about  $X_{\text{f}}\TeX$  – that's one of the irritating things about the Internet: You read about something in one place and then chase it down, in the process losing the original source of the information.  $X_{\text{f}}\TeX$  looked almost too good to be true: Unicode, and the ability to use both AAT and OTF fonts out of the box. Let me say that using  $X_{\text{f}}\TeX$  has been a transforming experience, and has re-energized my interest in  $\TeX$ . Thanks, Jonathan!

### 7.2 St. Will and the Miraculous fontspec

Last, but most certainly not least, is Will Robertson's wonderful fontspec package for  $X_{\text{f}}\LaTeX$ . Now, at this point I have to make a confession: I am a plain  $\TeX$  person. From my perspective,  $\LaTeX$  has become the Microsoft Word of the  $\TeX$  world. In the work that I have done, fonts are a major design issue, and the ability to change the typefaces quickly and efficiently in a given project or series of projects is an important requirement. Yes, I have spent hours with fontinst, afmtotfm, fd files and all the rest of it, but I do not consider it time well-spent. I would also point out that I have a love-hate relationship with Computer Modern: sometimes I hate it, at other times I love to hate it. Will's fontspec removes, for me, one of the major barriers to using  $\LaTeX$ , and for that I'm, well, if not exactly happy, at least pleurably intrigued.

So, at present, I think I can safely say that I am back to where I was when Textures was my  $\TeX$  tool of choice – indeed, I am much better off, with many more typographic options at my disposal.

## 8 ...Is Now...

Over the years, in addition to the book, I have done a tremendous amount of work for non-profit arts-oriented organizations (programs, pamphlets, order forms, survey forms and more), helped a friend to publish genealogical tables nicely formatted – all using plain T<sub>E</sub>X with Textures and more recently T<sub>E</sub>XShop. I’ve also burdened the CTAN archives with a series of type specimens collectively called *typespec*.

My time at the AMS has been... *interesting*, to say the least. Not only have I had to develop my unix skills (as those skills verged on the excessively exiguous, this was easy), but I’ve had to come to terms with something called VMS – ugh. At present we do all of our production work on VMS, though this is shortly to change: we hope to move everything to unix by the end of this year. This will be accompanied by a complete reorganization of the directory structure for processing our materials, and putting the whole thing under Subversion – yes, we’ve had a lot of fun with that – which will serve as both version control and online archive. We’ll have considerably more to say about this, I should think, at PracT<sub>E</sub>X 2007.

As if all of that were not enough, we have replaced our aging film imagesetter with a brand new platesetter from basysPrint of Boizenburg, Germany.<sup>3</sup> The interesting thing about this piece of equipment (not counting the fact that it is HUGE: seven feet square by five feet three inches in height, and weighing 2.5 tons!) is that it exposes standard, inexpensive, printing plates using a powerful source of ultraviolet light focused through a solid state chip<sup>4</sup> that contains thousands of tiny little mirrors that focus the light and create an image on the plate. The chip, alone, costs about \$9,000 to replace, should that ever be necessary.

## 9 ...And Ever Shall Be?

With all this, and my time at the AMS behind me, I would venture a few observations. Clearly, we are currently enduring the curse of living in “interesting times.” Whereas we used to live in a simple world divided between plain T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X, we now face an intimidating array of possibilities: T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X; pdfT<sub>E</sub>X and pdfL<sup>A</sup>T<sub>E</sub>X; X<sub>Y</sub>T<sub>E</sub>X and X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X; Context; Omega;... and so it goes. All this is wonderful for authors, but a

---

3. About 100km east of Hamburg.

4. Developed by Texas Instruments; also used in large-screen projection televisions.

veritable nightmare of sorts for publishers who have to streamline their production for maximum effect, minimum waste and quickest turnaround.

(By the way, what can be said for flavors of T<sub>E</sub>X goes double for graphics: the dizzying array of graphics packages (in many versions) that turn out a profusion graphics formats is becoming a real problem to publishers. But that is a subject for another day..)

At present, our production is based on L<sup>A</sup>T<sub>E</sub>X: for journals we require T<sub>E</sub>X files, and strongly encourage our authors to use the AMS class files (we typeset all of our journals inhouse). For books we require dvi files, though this has begun to shift slightly: If the author is willing to do *all* of the work to format the book to our standards, we will accept a PDF for the project. But how many times have we heard this at the AMS from an author: “I’m using T<sub>E</sub>Xshop which means I have to give you a PDF: T<sub>E</sub>Xshop can’t make a dvi file.” Hmmm... Or received PDF files with missing fonts, or PNG or jpeg files accompanying a T<sub>E</sub>X file?

And whereas at one time it looked as if PDF files might be a common denominator, now we need to ask which PDF specification – 1.1, 1.2, 1.3, 1.4, 1.5, or 1.6? Or PDFX-1a or PDFX-3... and the list goes on. Is the author using the Jaws PDF tools, or Ghostscript’s or Adobe’s? And not all PDF’s of the same specification are made equal, as we all know.

From time to time we get angry emails from exasperated authors (often from the physical sciences) demanding angrily why we are still using T<sub>E</sub>X, a tired, old-fashioned and awkward tool at best. Surely there is something else, more modern, more up-to-date that would be more suitable. Why not Microsoft Word? If I knew how to do a hollow laugh, I’d do one now.

Well, it’s a mess. In the end there is sure to be a catastrophe when increasing costs, diminishing income and tightening timeframes for publishers conflict with increasing pressure to publish as much as possible in the smallest period of time in the academic community. Working in the field of academic publishing right now is a bit like watching a train wreck happening in slow motion.

I’m not sure that I know the solution to these problems, though I can be quite certain that Microsoft Word is *not* the solution. So, I take myself as an example: I know my tools well enough to know that I can provide whatever a publisher requires of me. It is that simple. Publishers are in a different position than authors, because publishers have workflows that require a limited variety of input for them to produce reliable output. So it is my obligation as an author to provide what the publisher requires, if I want to be certain that I get what I want. But I’m lucky: I have wonderful tools at my disposal.

In large part, I think that the start of a solution is to put well-designed, versatile, and

comprehensible tools into the hands of authors, tools such as T<sub>E</sub>Xshop and i-Installer; and to provide documentation that is as good as the software. Not long ago we dealt with an author, a Mac user, who for the last ten years has been working on an important book, and using Textures to put it together. At some point he switched to OS X. As OS X developed and grew, Textures, sadly, did not. Finally the situation for him became intolerable and he asked the AMS for help. We pointed him in the direction of T<sub>E</sub>Xshop and i-Installer and gave him some pointers for getting started. In less than two weeks, the work of ten years was easily handed off to T<sub>E</sub>Xshop with no problems. Another transformative experience and a happy customer.

## 10 World Without End. Amen.

So it has been a long and never uninteresting road from 1986 to the present, with lots of challenges ahead. Yikes! Can it really be twenty years? In some ways it seems like yesterday, in others like a lifetime ago: Look at how the internet alone has changed our lives and our thinking in a mere fifteen years. To this day, I never cease to marvel at and be grateful for the intoxicating sense of empowerment that I have experienced with my Mac. I can only hope that everyone might experience the same joy with the tools that they use.

Thank you.

# A lifetime as an amateur compositor

David Walden

[THIS IS A PRELIMINARY DRAFT. THE CONTENT MAY CHANGE BY THE TIME OF THE PRAC $\TeX$ 06 CONFERENCE PRESENTATION AND/OR THE CONFERENCE PROCEEDINGS.]

*The first section of this paper briefly relates my experience writing and printing documents until I began to use  $\TeX$ . The second section summarizes why I now use  $\TeX$  and gives examples of its benefits, particularly writing books. Section 3 touches on the advantage of being able to use a separate powerful text editor, since  $\TeX$  does not require use of a built in editor. Go straight to section 2 if you want skip my reminiscences that are not directly related to  $\TeX$ .*

## 1 First fifty years

### 1.1 Pre-computers

For some reason, I have always been interested in putting marks on paper — as with many people, my first work was with crayons, finger paints, and 1 inch, ruled “primary paper” and thick “primary pencils.”<sup>1</sup> But it was not long before I moved on to more publication-like processes. Our church had a mimeograph machine and my parents were involved with producing the Sunday programs, and my parents both taught in the public schools<sup>2</sup> where they prepared handouts to students using ditto machines. I was a little involved with reproducing such materials at least throughout my teen years.

My father had an Underwood manual typewriter upon which he typed and on which I banged with a few fingers as a child. Later, he obtained a Royal manual typewriter on which I typed with ten fingers from the time I took a typing class two hours a week for one term during my sophomore year of high school. Ever since taking that typing class, I have been typing, often for reproduction, more or less constantly: so much so that when my son was a child and people asked him what his father did, my son’s answer was, “He is a typist.” After I went away to college, I moved to a Smith Corona electric portable typewriter; and when I entered the work

<sup>1</sup> The oral presentation version of this paper given at the Prac $\TeX$ 06 conference included a number of photographs that are not included here because I did not seek permission to use them.

<sup>2</sup> Supported by the town government.

world, we used IBM Selectric typewriters with their changeable type balls.

However, I wasn’t a mistake-free typist, and I had much use for the tools of typewriter correction using carbon paper and other typewriter-base media: erasers, stuff to patch a mimeograph stencil, a razor blade to scrape the ink off of a ditto master, and KO-REC-TYPE paper and Snopake correction fluid to paint over typing so characters could be re-typed correctly on pages that would be reproduced on Xerox copiers.

At <http://www.tpub-products.com/>, I found a document for sale that describes the duties of a military “religious program (RP) specialist” (an assistant or secretary to a chaplain), and it includes instructions for using Ditto masters; I quote it below. This description represents about the mid-level of complexity of pre-computer desktop “publishing” — more complicated than carbon paper (but not much) and slightly less complicated than a stencil machine.

Before proceeding to an explanation of stencil preparation, the Ditto master will be discussed. The white Ditto master (overlay) is attached to a sheet of paper which is thickly coated with a carbon substance. Typing and hand- stylus impressions are made on the overlay and cause the carbon substance to be imprinted on the reverse side of the master. When the overlay is attached to the Ditto machine, the carbon-coated sheet is detached. The carbon impressions of the Ditto master are moistened by the duplicating fluid as the drum is rotated, which in turn transfers the carbon dye to the paper being fed into the machine. This transfer yields an exact reproduction of the master.

Preparing a neat and accurate Ditto master stencil is one of the more important secretarial tasks that the RP will perform. Command Religious Program announcements are often distributed to command personnel through the use of Ditto copies. Just as the appearance of the office of the chaplain makes an instant and lasting impression, an information “flyer” or announcement will also leave lasting impressions. If the announcement is neatly prepared with concise and accurate information, it will probably give people the impression that the office of the chaplain is an efficient and caring organization. Therefore, it is important that the RP prepare each Ditto



master with these thoughts in mind. The following helpful hints should aid the RP in preparing Ditto masters:

- The “flimsy” sheet of paper that is inserted between the Ditto overlay and the carbon attachment **MUST** be removed before it is possible to have impressions transferred to the back of the overlay. **NOTE:** If there is some art-work involved, the “flimsy” may be left between the overlay and carbon attachment while the art-work is penciled lightly onto the overlay. The art-work can then be retraced with a stylus when the “flimsy” is removed. If an electric typewriter is being used, a test line should be typed on a Ditto master at each typing pressure setting. A copy should then be run and the RP can select the pressure that will provide the best copy. For manual typewriters, the typing pressure lever should be set to a medium or light position for best results.

- A Ditto master should be left in the typewriter when errors are corrected. The typewriter platen should be turned until there is enough room to separate the perforated overlay from the carbon backing. A razor blade or other sharp-edged instrument should then be used to lightly scrape the carbon deposit of the incorrect characters from the back of the overlay. Next, a clean piece of Ditto carbon should be placed between the overlay and the original carbon. Then the typewriter should be returned to its original position and the correct letters typed. After the correction has been made, the temporary carbon that was used for this correction **MUST** be removed before proceeding.

- Ditto masters may be reused at a later date if they are properly stored after the initial use. The masters should be placed in large envelopes and separated by flimsy sheets. It is imperative that they be stored in a flat position to keep them from becoming wrinkled

The point I am trying to make with the above discussion about the pre-computer era is that it took many (fussy, touchy, tedious) steps of careful work to produce good output, just as it does today in the world of fancy computer-based systems. Added problems were that the “desktop” (versus professional printing) approaches to reproducible typesetting in the pre-computer era didn’t produce high quality printing, and there were limits on the number of copies you could make before the masters wore out.

## 1.2 Early computer use

I first came in contact with computers when I was in my junior year in college. While I still continued to use a typewriter for the next decade or so, I was also phasing over to using computers for typing documents, especially those that would be repro-

duced. I started with punch cards and an IBM 025 key punch machine, moved to rolls of punched paper tape with editing using Dan Murphy’s TECO (tape editor and corrector), continued using TECO (modified to work with computer files rather than paper tape) via a Model 33 Teletype and then a TI Silent 700 as I moved into the world of computer time sharing (where the computer terminal was in my own office for the first time), used Jeremy Salter’s RUNOFF (the first word processing program) on MIT’s CTSS system, MRUNOFF (a version of RUNOFF for the TENEX operating system), and briefly touched troff/nroff in the early years of UNIX. This computer-based world allow editing (e.g., with TECO) and reprinting of the actual raw text of a document or, eventually, inclusion of typesetting commands that would be interpreted by RUNOFF, MRUNOFF, and troff/nroff to produce the final document which could then be reproduced.

In the mid- to late-1970s, I first used a personal computer — an Apple II — but only to run VisiCalc. I was still doing word processing using MRUNOFF on TENEX. In 1981, IBM announces its PC and I got one for the following Christmas, I believe. My wife began using WordStar, and I helped her because I was familiar with command-based word processing from MRUNOFF which my friend Rob Barnaby (developer of WordStar) had also used.

## 1.3 Word and WYSIWYG

I don’t remember when, but before too long (perhaps on the first PC AT) I began using the WYSIWYG PC-Word for DOS (based on the ideas of Charles Simonyi). Then I converted to using the Mac and MacWord which seemed to be where the forefront of Word development was taking place. MacWord was somewhat incompatible with PC-Word, but my PC-Word files converted over the MacWord pretty well, although my memory is that the very straight forward style sheets of Word for DOS were no longer quite so straight forward with MacWord and I couldn’t find some other features I had been using with PC-Word. I used MacWord for about eight years. In the early 1990s I decided to convert back to using an IBM PC using a Windows-based DOS operating system and then Win 3.1, but I discovered that my original Word files for the early PC that had been converted to MacWord did not convert back to the later versions of PC-Word version well. This was quite distressing to me. Moreover, after each of these changes I could not find various capabilities I was used to using — they were perhaps still there but apparently had moved or how they were executed had changed.

As time went by and I continued to use Word as part of Microsoft’s Office Suite, I became increasingly annoyed at Word. Bigger, more complicated releases kept coming out, and in time there was pressure from people with whom I exchanged Word files to upgrade to the latest version because earlier versions couldn’t easily handle files from later versions without the person using the later version explicitly saving the files in the format of the earlier version, something many Word users didn’t even know how to do. Also, each new release tended to again change how one called for various capabilities to be executed, while in time Microsoft stopped shipping hard copy manuals with Word from which one could learn such things (Microsoft increasingly forced users to depend on on-line documentation which doesn’t work so well when you don’t know how to ask for what you want to know about). Also, each new release tended to try to do more things automatically for me, and it took more and more work to turn off all the “help” it was trying to provide to me—help that in many cases actually made things harder for me (while it didn’t help me by providing powerful editing functions, e.g., using regular expressions).

#### 1.4 Breaking with Word; choosing T<sub>E</sub>X

My level of annoyance and frustration grew and grew, and eventually I made the decision to stop using Word for significant writing projects and to seek an alternative. Before I go on about my alternate approach, I must emphasize that I still use Word regularly for short, one-off projects (e.g., a short letter that I will not need to access on-line at a significantly later time) and when I work with someone who uses Word for his or her document preparation.

I chose to use L<sup>A</sup>T<sub>E</sub>X as my alternative to using Word for document preparation for several reasons:

- It had a visible, non-proprietary, documented markup with a simple, plain text syntax that I was confident would allow me to reuse text in different documents over the years.<sup>3</sup>
- I was already familiar with command-based word processing and (as a computer programmer at heart) liked what I know about T<sub>E</sub>X’s programmability. I also welcomed the prospect of being able to use a powerful text editor again as part of my document editing process (more about this in section 3).

<sup>3</sup> Word’s hidden markup and WYSIWYG editing means that it is often hard to tell how something got to be the way it is. Also, since many Word users don’t use style sheets, formatting (for instance, of a subsection title) might be done one way for one subsection and another way for another subsection, increasing the probability of inconsistencies in the output.

- I had been involved with religious arguments about which of PageMaker, FrameMaker, or Interleaf was the best tool in various situations; and, from what I knew then, they also had some of the same problems as Word in terms of hidden markup and pressure on users to adopt new releases that obsoleted prior releases. I also was definitely looking for something that did not involve a graphical user interface (GUI)—something that required less mouse clicks. So, I didn’t seriously investigate the just mentioned systems.
- I am a great admirer of Donald Knuth and thought it would be nice to try the system he developed.

Part of my preference for L<sup>A</sup>T<sub>E</sub>X over Word comes from the fact that all of the markup is in a file where I can see it and change it rather than it having to be accessed by various menus and being largely unseen (except in its effect) in the document. To take a simple example, suppose I wanted to make the word “brown” in the phrase “quick brown fox” be bold. In Word I would select “brown” with the mouse, pull down the Format menu, click the Font item on the menu, click Bold in the Font Style column, click OK, and then the text would appear in the document in bold when displayed or printed (alternatively I could type control-B after selecting the word “brown”). To do the same thing in L<sup>A</sup>T<sub>E</sub>X, I would change the text “quick brown fox” to “quick `\textbf{brown}` fox” with my text editor, and “brown” would display in bold when my L<sup>A</sup>T<sub>E</sub>X file was compiled.

No doubt there are ways in Word to do many if not all of the things I now do with L<sup>A</sup>T<sub>E</sub>X, but I find them mostly easier to find and do in L<sup>A</sup>T<sub>E</sub>X.

As an aside, another aspect of Word that annoys me is that it is forever guessing what I want. For instance, if I type an explicit new-line (Return key), Word may decide to capitalize the first word of the next line, which may or may not be right. When I select some text with the mouse in Word, it often chooses different text than I touch with the mouse, for instance an extra space. Much or all of this can probably be turned off and I turn off as much as I can, but I never seem to be able to turn off everything; and, while Word’s “help” sometimes does result in what I want, it seems more often to choose what I don’t want. L<sup>A</sup>T<sub>E</sub>X never seems to cause me this problem, which is not to say there are not other problems with L<sup>A</sup>T<sub>E</sub>X.

I don’t remember what T<sub>E</sub>X distribution—I downloaded something from the Internet—I tried first using NotePad on the PC for my editing. I do remember buying *The T<sub>E</sub>Xbook*, and then quickly

discovering L<sup>A</sup>T<sub>E</sub>X which I experimented with a little bit. Then, I bought a copy of PCT<sub>E</sub>X on the theory that it would be nicely packaged, and I used it for a while but grew dissatisfied with the power of its editor. Then I found and downloaded WinEdt and MiK<sub>T</sub><sub>E</sub>X. Later I bought and tried the Y&Y distribution, but I could never get it to work well; I did buy and make good use of the V<sub>T</sub><sub>E</sub>X distribution for one particular project, but again I didn't like its editor. I ended up using WinEdt (and occasionally EMACS for things that seemed harder to do in WinEdt than in EMACS) and MiK<sub>T</sub><sub>E</sub>X for a number of years, most recently obtained as part of TUG's Pro<sub>T</sub><sub>E</sub>Xt distribution.

## 2 Why I use L<sup>A</sup>T<sub>E</sub>X, particularly for writing books

Two reasons typically given for using L<sup>A</sup>T<sub>E</sub>X are for its math support and for very nice looking typesetting. Neither of these is particularly important to me: I rarely have any math in my writing (but it is nice to be able to handle it easily in those rare cases where I do have it); I have a pretty undiscerning eye when it comes to typesetting, and what L<sup>A</sup>T<sub>E</sub>X produces is more than good enough for me.

The things that matter most to me about L<sup>A</sup>T<sub>E</sub>X are:

1. its programmability and modularity
2. that I get to use a powerful editor with it
3. that the mark-up is clearly visible to me and can be changed directly with a text editor
4. its capabilities for explicitly specifying cross-references, maintaining bibliographies, and automatically numbering chapters, sections, figures, tables, footnotes, etc., which permit easy reorganization of text within documents and reuse in other documents
5. its relatively slow pace of change and great concern among the developers for backwards compatibility

In other words, my use of L<sup>A</sup>T<sub>E</sub>X is primarily about productivity. (Of course, there are certain limitations on this productivity such as when I finish writing a book using L<sup>A</sup>T<sub>E</sub>X and the publisher tells me I must convert the text to Word and the figures to PowerPoint slides for input into the compositor's typesetting system.)

Much of my work using L<sup>A</sup>T<sub>E</sub>X is on book length documents. For these I have compiled a more or less standard set of techniques that I feel help me be more efficient. I don't claim that the techniques I use are the techniques of a master; in fact, I view myself as an intermediate user of L<sup>A</sup>T<sub>E</sub>X — I know

enough to make L<sup>A</sup>T<sub>E</sub>X jump through a few simple hoops, but not enough to know if my approaches are recommended or if they include some bad habits.

In my experience, publishers don't think much about the design of a book until they have the completed manuscript in hand. Since I use L<sup>A</sup>T<sub>E</sub>X to develop the original manuscript, I have to make lots of temporary design decisions, and I want to be able to change these decisions with a minimum of work when the publisher does begin to deal with the design. Also, I am currently working on a book that I will be self publishing, and settling on the design for this book is an iterative, experimental process where it is even more important to be able to make changes throughout the book (for instance, to the style of figure captions) with minimal work. My experience, however, should not prevent you from checking if the publisher of your document already has a standard style and perhaps even a L<sup>A</sup>T<sub>E</sub>X class file that you can use from the the outset of your writing. In any case, my emphasis here is *not* on the methods of representing preferences for appearance; my emphasis is on methods for easily and repeatedly *changing* the overall document appearance as well as on other methods for working efficiently on large documents.

Some of what I am about to describe for working efficiently on books or other long documents is probably already well known to many readers; perhaps you can make suggestions for how I might do things better.

(At several points in the following, I have included in parentheses discussions of basic T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X issues that reviewers and others who have read drafts of this paper have asked me about that are not actually on the subject of book-writing productivity. Perhaps these parenthetical notes should have been footnotes, but I was too lazy to deal with the need for alternatives to `\verb` in footnotes.)

### 2.1 Include files

Suppose I am working on a book entitled *Breakthrough Management*, as I have been recently. I created a top level file named `bt.tex` with the following contents:

```
\documentclass{btbook}
\begin{document}
\include{titlepages}
\include{preface}
\include{surviving} % a chapter
\include{rapid}    % another chapter
. . .              % more chapters
\include{acknowledgements}
\include{bibliography}
\include{bio}
```

```
\include{index}
\end{document}
```

The text from included files appears to L<sup>A</sup>T<sub>E</sub>X as if it was in the file `bt.tex` in place of the `\include` commands. In this way, I contain the text related to each chapter and other parts of the book in its own file. I let L<sup>A</sup>T<sub>E</sub>X take care of numbering the chapters and figures (or whatever) within chapters. If I later decided to change the order of chapters, I just change the order of the `\include` commands in the `bt.tex` files, and L<sup>A</sup>T<sub>E</sub>X automatically rennumbers everything.

To work on one chapter at a time, my file `bt.tex` evolved to include many `\includeonly` commands, e.g.,

```
\documentclass{btbook}
%\includeonly{preface}
%\includeonly{surviving}
\includeonly{rapid}
%\includeonly{surviving,rapid}
. . .
\begin{document}
\include{titlepages}
\include{preface}
\include{surviving} % a chapter
\include{rapid}    % another chapter
. . .              % more chapters
\include{acknowledgements}
\include{bibliography}
\include{bio}
\include{index}
\end{document}
```

In the above example, only the file `rapid.tex` gets compiled when I run L<sup>A</sup>T<sub>E</sub>X on the file `bt.tex`. In this 10 chapter book I had a couple of dozen `\includeonly` commands in the `bt.tex` file that I could comment in and out to work on each chapter individually and with various combinations of related chapters.

(Because the `\include` commands result in text being typeset, they must follow the `\begin{document}` command. The `\includeonly` commands must go in the preamble or else L<sup>A</sup>T<sub>E</sub>X complains.)

## 2.2 Custom class file

I have created a file `btbook.cls` which is my own personal class file for this particular book. This file is processed when L<sup>A</sup>T<sub>E</sub>X sees, at the beginning of the file `bt.tex`, the command `\documentclass{btbook}`. The first three lines of the file

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesClass{btbook}[2006/01/21 BTbookclass]
\LoadClass{book}
```

define the class for this book to be named `btbook` and to be an augmentation of the L<sup>A</sup>T<sub>E</sub>X book class.

The rest of the lines of the file are read and executed when L<sup>A</sup>T<sub>E</sub>X is run as if they were lines of text immediately

following the `\documentclass` command in the `bt.tex` file.

(If your publisher already provides a L<sup>A</sup>T<sub>E</sub>X class file, you can still collect all of the sorts of things I describe below in their own file and `\input` that file in the preamble rather than just putting all these things directly in your preamble. I prefer not to have much in my preamble beyond the `\includeonly{...}` commands that I am constantly commenting in and out.)

## 2.3 Packages

Next in the class file come the list of packages I use for writing this book.

```
% Palatino is basic roman font
\RequirePackage{mathpazo}
% Helvetica is sans serif font
\RequirePackage[scaled=.95]{helvet}
% Courier is typewriter font
\RequirePackage{courier}
% for including images
\RequirePackage{graphicx}
% for formatting URLs
\RequirePackage{url}
%to be able to rotate figures
\RequirePackage[figuresright]{rotating}
% for dropped caps
\RequirePackage{lettrine}
% for tighter list spacing
\RequirePackage{paralist}
\setlength{\pltopsep}{.05in}
% for comment environment
\RequirePackage{comment}
% for endnotes with reformatted numbers
\RequirePackage{dw-endnotes}
%\doublespacing
\RequirePackage{setspace}
```

When I find I need to use another package, I add another `\RequirePackage` line to this list. (As I understand it, `\RequirePackage` does the same job as `\usepackage` except it doesn't allow the same package to be loaded twice which apparently might cause problems in some cases.)

Notice that the package name in one case includes the characters `dw-`. This is my convention for noting a package that I have modified. In such cases, the file of the modified package is in the same directory with the rest of the files for this book or in the local changes part of my `texmf` data structure. I seldom understand a package I am modifying; I typically use a hit and miss approach to change stuff until I get the results I want.

Copy editors who edit on hard copy like double spacing, and I can provide that with a one character change—uncommenting the `\doublespacing` command on the last line above that loads the `setspace` package.

## 2.4 Miscellaneous useful macros

The following macros provide a few capabilities I use relatively frequently.

```
% space around em-dashes
```

```

\newcommand{\Dash}{\thinspace---\thinspace}
% mark text that needs checking
\newcommand{\CK}[1]{\textbf{CK #1}}
% marginal note to myself
\newcommand{\manote}[1]{%
\marginpar{\scriptsize To do:\#1}}
% cut-in title
\newcommand{\partitle}[1]{%
\medskip\noindent\textbf{#1}}
%change function of \url command
\let\Originalurl=\url
\def\url#1{\fontsize{10.7pt}{13.05pt}
\Originalurl{#1}}

```

For some documents I have worked on, I have had many more such miscellaneous useful macros.

Anyone trying to improve productivity using L<sup>A</sup>T<sub>E</sub>X who doesn't already define his or her own macros should learn to do so. User-defined macros allow significant improvements in efficiency. For instance, the first macro above defines the command `\Dash` to be an abbreviation for the character string `\thinspace---\thinspace` which results in an em-dash being typeset with a *little* bit of space on each side of it, as in `aaa—bbb`. It is less characters and probably more reliable to type `\Dash` many times in a book than it is to type the characters `\thinspace---\thinspace` many times. In my view, however, the greater benefit of defining the `\Dash` command comes when my publisher tells me that its style is closed-form em-dashes (no space on each side, i.e., `aaa—bbb`) or a more open form (`aaa — bbb`). To implement either of these changes *throughout* the book, I merely redefine `\Dash`, e.g.,

```

% no spaces around em-dashes
\newcommand{\Dash}{---}

```

or

```

% full spaces around em-dashes
\newcommand{\Dash}{ --- }

```

and recompile my document. Containing such style conventions within a few lines of a large document and being able to change the style throughout the document with only a few key strokes is an enormous advantage. (I'll give a more complex example of such containment when I discuss macros for figures and tables below.)

To redefine a command that already exists in L<sup>A</sup>T<sub>E</sub>X or has been defined by a package that has already been loaded, for instance to define a variation on `\url` as I do in the last two lines of my group of miscellaneous useful macros, I have to use the `\renewcommand` command. The `\renewcommand` works just like `\newcommand` except that L<sup>A</sup>T<sub>E</sub>X doesn't complain with `\renewcommand` if I try to give a definition to a command that already exists—a good thing to be warned about when one uses `\newcommand`. (In the next subsection I give another redefinition example—redefining `\footnote`).

## 2.5 Footnotes and endnotes

In the case of `\RequirePackage{dw-endnotes}`, I am using the `endnotes` package, modified slightly to change the format of the note numbers.

Typically, I put footnotes on the bottom of text pages where they are referenced, at least while I am drafting chapters and want to be able to see the notes without having to turn a bunch of pages. However, publishers tend not to like having footnotes—it makes a book look too academic to be popular, in their view. Thus, before actually publication, I often find myself converting all my footnotes to endnotes. The next commands in my class file do this.

```

%comment out to not have end notes
\renewcommand{\footnote}{\endnote}

\newcommand{\dumpendnotes}{%
\medskip
\begingroup
\setlength{\parindent}{0pt}%
\setlength{\parskip}{1ex}%
\renewcommand{\enotesize}{\normalsize}%
\theendnotes
\endgroup
\setcounter{endnote}{0}}

```

First, the `\footnote` command is redefined to be the `\endnote` command; this avoids me having to replace every instance of `\footnote` with `\endnote`. Then the class file defines a command (`\dumpendnotes`) that can go at the end of each chapter to dump the chapter's endnotes, formatted as I want them to be. If the command `\dumpendnotes` was already defined in L<sup>A</sup>T<sub>E</sub>X or some other package, L<sup>A</sup>T<sub>E</sub>X would warn me because I didn't do the definition with `\renewcommand`.

## 2.6 Formatting figures and tables

The next set of commands in the class file have to do with changing the format of figure and table captions without actually modifying a L<sup>A</sup>T<sub>E</sub>X or package file. The L<sup>A</sup>T<sub>E</sub>X default does not use bold face for captions and uses a period rather than a hyphen between the the chapter number and figure number within a chapter. The following changes patch L<sup>A</sup>T<sub>E</sub>X to follow my preference for bold face and hyphens.

```

\long\def\@makecaption#1#2{%
\vskip\abovecaptionskip
\sbox\@tempboxa{\textbf{#1}. \textbf{#2}}%
\ifdim \wd\@tempboxa >\hsize
{\textbf{#1}. \textbf{#2}\par}
\else
\global \@minipagefalse
\hbext@\hsize{\hfil\box\@tempboxa\hfil}%
\fi
\vskip\belowcaptionskip}
\renewcommand \thefigure
{\ifnum \c@chapter > \z@ \mbox{\thechapter-}
\fi \@arabic\c@figure}}

```

```
\renewcommand \thetable
{\ifnum \c@chapter>\z@ \mbox{\thechapter-%
\fi \@arabic\c@table}}
```

(I do not have to bracket these lines, top and bottom, with `\makeatletter` and `\makeatother` commands as I would have to if this patch was in the preamble of my document; the at-sign is a letter by default in class files and packages. Some readers may be back a step, at the question of, “What is it about at-signs anyway?”: The answer is that the files for basic L<sup>A</sup>T<sub>E</sub>X, for class files, and for other packages are full of macros names that include an at-sign (@), e.g., a macro named `\@makecaption` is defined at the beginning of the above example. My understanding is that an at-sign is used in low level programming of L<sup>A</sup>T<sub>E</sub>X, class files, and packages to create macro names that can’t accidentally conflict with names that may be defined by users not doing such L<sup>A</sup>T<sub>E</sub>X “systems programming.” An at-sign is normally not a letter and thus cannot be part of a macro name. However, in the above example I want to patch low level L<sup>A</sup>T<sub>E</sub>X code that includes at-signs in its macro names; if I was trying to make this patch in my preamble (as I used to do before I learned to make some patches in a personal class file), I would have to tell L<sup>A</sup>T<sub>E</sub>X to temporarily turn at-signs into letters, make the patch, and then turn them back into non letters (other) so the rest of my program could use @ in the normal way where it is not a special character of any kind.)

Perhaps there is a caption package that would allow such changes without patching L<sup>A</sup>T<sub>E</sub>X, but I was shown how to make this patch a few years ago and it works, so why bother trying to find and learn a new package?

Next in my class file come a set of definitions for commands I use to include graphics. I seldom insert `\begin{figure}` and `\end{figure}` commands directly into my documents; I do, from time to time, insert the commands `\begin{table}` and `\end{table}`. It is inevitable that, before I am done with a big document, I will want to change the formatting relating all figure and tables — perhaps several times. Thus, I use macros for inserting almost all figures (or tables) such that I can make changes to formatting relating to the figures by making changes to only a few lines in the relevant macros.

```
%switch argument among pdf, eps, etc.
\newcommand{\figfiletype}{pdf}
%tell LaTeX directory path to figures
\graphicspath{{figures/}}
%commands to display file name, or not
\newcommand{\DFN}[2]{%
\texttt{\small[#1 #2]}}
%\newcommand{\DFN}[2]{}

\newcommand{\snfig}[3]{%scaled numbered figure
%drop htb and %s for single page figures
\begin{figure}[htbp]
%\vbox to \vsize{%
\hfil\scalebox{#3}{
```

```
\includegraphics{#2.\figfiletype}}\hfil
\caption{\label{fig:#2}#1 \DFN{#2}{#3}}
%\vfil
%}
\end{figure}
}

\newcommand{\sntab}[3]{%scaled numbered tables
\begin{table}[htbp]
%\vbox to \vsize{%
\centering
\caption{\label{tab:#2}#1 \DFN{#2}{#3}}
\smallskip
\scalebox{#3}{
\includegraphics{#2.\figfiletype}}
\label{tab:#2}
%\vfil
%}
\end{table}
}

\newcommand{\unfig}[2]{%scaled unnumbered fig.
\begin{figure}[htbp]
\hfil\scalebox{#2}{
\includegraphics{#1.\figfiletype}}\hfil
\label{fig:#1}\centerline{
\DFN{#1}{#2}}
\end{figure}
}

%sideways scaled numbered figure
\newcommand{\swsnfig}[3]{
\begin{sidewaysfigure}
\centering
\scalebox{#3}{
\includegraphics{#2.\figfiletype}}
\caption{\label{fig:#2}#1 \DFN{#2}{#3}}
\end{sidewaysfigure}
}

For instance, the macro \snfig above takes three arguments. The text for a figure caption, the unique part of the file name for the graphic to be included, and a scale factor for the graphic, e.g.,

\snfig{This is the caption}{figure3-31}{.8}
```

The full name of the file to be included is the concatenation of the part of the file name that came from the second argument of the macro call, the directory that is specified by the `\graphicspath` command (an option of the `\graphicx` package) as the place L<sup>A</sup>T<sub>E</sub>X searches for figures, and the `\figfiletype` definition as the file name extension. The latter is useful because sometimes all of my figures are .eps files and sometimes they are .pdf files, and sometimes I switch between these two formats at different times in the production of the book. (When using .eps format, I compile using L<sup>A</sup>T<sub>E</sub>X and a dvi-to-pdf conversion; when using .pdf format, I use pdfL<sup>A</sup>T<sub>E</sub>X to compile. If the graphic format was changing from

file to file within the document, I would instead specify the format as another argument to the `\snfig` command. [However, Will Robertson has recently pointed out to me that if I leave the extension off, `\includegraphics` will pick the appropriate extension: `.eps` for  $\LaTeX$  and `.pdf` for  $\pdf\LaTeX$ .]

While I am drafting and revising a book manuscript, I want to be able to look at a figure in the printed output and know what file I need to modify to change the figure. Thus, my macros for including figures and tables causes the file name to be included in the printed output in small letters enclosed in small square brackets, using the macro `\DFN`. When it comes time to create the final manuscript, I swap to a definition of `\DFN` that produces nothing and recompiles the book's  $\LaTeX$  files.

The definitions of `\snfig` and `\sntab` also include several lines that are commented out. Professional editors often like to see the manuscript with figures or tables each on its own page rather than in-line with the text. Commenting in these few lines puts the figures and tables of the whole book on their own pages.

The `\snfig`, `\sntab`, and `\swsnfig` macros also define labels for cross referencing the figures with `\ref` or `\pageref` commands. A slight limitation of my implementation is that I cannot reuse the same figure or table file without confusing the labeling. However, it is easy enough to create a duplicate figure or table with a different file name.

I typically create all figures and most tables outside of  $\TeX$  itself and include them from separate files. If I found myself inserting very many tables directly into my `.tex` files rather than including them from graphics files, I would define a `mytable` environment so that I could still contain and simply change the sort of formatting I have discussed.

## 2.7 Thought breaks

The next group of commands (mostly commented out) are various options for indicating what I call “thought breaks” — places where formatting indicates a change of topic big enough to highlight but not big enough to have its own section or subsection title.<sup>4</sup> (These commands are defined with `\def` because I know they will pick up the correct arguments this way, and I am not sure enough of the details of how `\newcommand` works. I understand the details of how  $\TeX$  defines a macro and then collects its arguments when the macros are called because Knuth explains it pretty completely in *The  $\TeX$  book*. In particular,  $\TeX$  allows macro calls where the arguments of the macro are not all embedded in pairs of braces. However, I have never stumbled across a rigorous explanation of how a macro defined with `\newcommand` collects its arguments and thus in what situations arguments not in braces will be recognized or to what extent  $\LaTeX$  defined macros can have both of what Knuth calls delimited and unde-

<sup>4</sup> See my *PracTeX Journal* 2005-4 “Travels in  $\TeX$  Land” column (<http://www.tug.org/pracjourn/2005-4/walden/>) for examples of thought breaks.

limited arguments — and I have not bothered to study the  $\LaTeX$  code to figure it out. Consequently, out of ignorance, I use `\def` to define macros which don't have their arguments delimited by braces.)

```
\begin{comment}
\def\newthoughtgroup#1{\bgroup
\afterassignment\BigFirstLetter \let\next=}
\def\BigFirstLetter#1{
\bigskip\noindex{\Large#1}}

%adapted slightly from Victor Eijkhout on ctt
\def\newthoughtgroup#1{\BigFirstLetter#1$}
\def\BigFirstLetter#1#2${
\bigskip\noindent{\Large #1}#2}
\end{comment}

\def\newthoughtgroup#1{%
\bigskip\noindent {\large #1}}

\begin{comment}
\def\newthoughtgroup{%
\bigskip\noindent }

%big bold dropped cap letter with rest
% of word small caps
\def\newthoughtgroup#1#2 {
\bigskip\noindent\lettrine{#1}{#2}\ }

\def\thoughtbreak{\vskip2pt
\centerline{$^{\vrule width2cm height 1pt}$}
\vskip2pt\noindent}
\end{comment}
```

The version of `\mythoughtgroup` currently not commented out indicates the new thought by a vertical space and a slightly bigger capital letter at the beginning of a non-indented paragraph.

## 2.8 Chapter formatting

The final set of commands in my class file has to do with the beginning and ends of chapters. At the beginning and ending of each chapter I insert some commands that I can change either by changing the commands themselves or changing macros in the class file.<sup>5</sup>

```
\RequirePackage{fancyhdr}\pagestyle{fancyplain}
\newcommand{\mypartname}{}
\newcommand{\mychaptername}{}
\lhead[\fancyplain]{
\thepage}]{\fancyplain}{}
\chead[\fancyplain]{
\mypartname}]{\fancyplain}{\mychaptername}
\cfoot[\fancyplain}]{
\fancyplain{\thepage}{}}
```

<sup>5</sup> When I first started customizing my page headings a few years ago, I used the `fancyheadings` package; recently I have learned that the package `fancyhdr` has replaced `fancyheadings`, but I have not yet bothered to rewrite all the heading commands to use the new forms that come with the `fancyhdr` package and don't use the `fancyplain` device.

```
\rhead[\fancyplain{}]{ }{
  \fancyplain{}{\thepage}}
\newcommand{\EMPTYPAGE}{\clearpage
  \thispagestyle{empty}\cleardoublepage}

\newcommand{\ENDCHAPTER}{\dumpendnotes}
\newcommand{\fENDCHAPTER}{\vfil\dumpendnotes}
```

In the class file for the book from which I drew these illustrations, there are a couple of alternative macros that I can include at the end of each chapter to dump the endnotes, but the end-of-chapter macros could be defined to cause other actions and outputs. In this book (which has only 10 chapters) I do not combine everything in a single beginning-of-chapter macro (e.g., `\BEGINCHAPTER`), but I have done this with some books (e.g., the 20 chapter book I am also currently working on). The typical beginning-of-chapter commands for the chapter with the file name `rapid.tex` (mentioned earlier) are

```
\EMPTYPAGE
\chapter{Rapid Change in a Global World}
\label{ch:rapid}
\renewcommand{\mychaptername}{Chapter %
  \thechapter: Rapid Change in a Global World}
```

### 2.9 Using a fully developed class

For some books I have included different or additional capabilities in my custom class file.

Obviously, I could also use a fully developed class such as memoir rather than making lots of modifications of my own to the L<sup>A</sup>T<sub>E</sub>X’s standard book class. However, I suspect I would still use some of the ideas I have described above.

It is clear that T<sub>E</sub>X and its derivatives with their explicit, visible markup provide a strong base for incrementally building a personal library of techniques that are easy to apply from one project to the next.

## 3 The possible benefits of a separate editor

Using a word processor such as MS Word that has invisible, undocumented, proprietary markup means you have to use its built-in, WYSIWYG editor that knows about that markup. This has two potential disadvantages: (1) GUI-based editing often takes a lot more key strokes to do simple things than an editor like WinEdt or EMACS (I gave an example of this in section 1 and provide additional examples below); (2) an editor like MS Word’s does not seem to have a lot of useful features that an editor like WinEdt or EMACS has.

Many of the ideas in this section are probably relevant to any typesetting system that has editing capability (like those I describe below). For all I know, MS Word can do many of these things; but, as I said in the first section, a number of years ago I lost interest in struggling to find stuff buried in all Word’s menus, dealing with its ever changing user interface, and its planned-obsolence-and-forced-upgrades business strategy.

### 3.1 Two ways to make a change throughout a document

In the last section I sketched the benefits of using macros for some sequences of chapters (for instance, `\Dash{}` for `---`) that enable the replacement sequence to be changed everywhere in a document by just changing the definition of the macro in one place. Another option for making a change to the same sequence of characters throughout a document is to use a text editor’s Replace All command. For instance, suppose I hadn’t used a macro for em-dashes and instead had closed form instances of `---` throughout my document, e.g., “this is the end—the end of the line.” And then suppose I decide to change the style to uses semi-open form em-dashes, e.g., “this is the end—the end of the line.” With my editor I can do a Replace All of `---` by `\thinspace---``\thinspace{}`. If the document is broken up into separate files for each chapter, it will be good if the text editor has the option for doing the Replace All over all documents open in the editor instead of only in the document where the cursor currently is.

Here is another example of a simple text replacement of the entire document. Suppose I decide (for some reason) to replace all en-dashes by hyphens. Then I can do the following sequence of three steps (the first and third steps are to avoid accidentally changing instances of `---` into `--`):

```
Replace All --- with #X#X#
Replace All -- with -
Replace All #X#X# with ---
```

Now suppose I want to add a fourth argument to every instance of the macro call `\snfig{ }{ }{ }` (see definition and discussion of this macro in subsection 2.6), that is, change the macro call formats to `\snfig{ }{ }{ }{ }`. Of course, one approach is to search for each instance of `\snfig{ }`, then move the cursor to after the third pair of braces, and then type the fourth pair of braces. However, if your text editor has a capability for dealing with regular expressions, you can make this change more easily (the last book I wrote had a couple of hundred instances of `\snfig`).

While I won’t get into the specific format of any particular editor’s representation of regular expressions, they would do something like the following in our example case.

```
Replace All \snfig{(.*)}{(.*)}{(.*)}
with \snfig{ $\$1$ }{ $\$2$ }{ $\$3$ }{ }
```

Everything in the Replace All command that is in a typewriter format is literal characters to be replaced. The characters in italics in the first part of the command are special characters that match any number of characters between balanced braces. For instance, in the command `\snfig{Caption title.}{file-name}{.5}` the first instance of `(.*)` would match `Caption title.` The second instance of `(.*)` would match `file-name`, and the third instance would match `.5`. Better than that, the editor stores each matched set of characters in its



own place for later reuse, as in the second half of the above command. The part of the command after the word “with” says to replace the `\snfig` command that was matched with all the same literal characters for the command names and braces, but to put the first match text in place of `$1`, the second match text in place of `$2`, and the third matched text in place of `$3`, and to add an extra pair of literal braces at the end of the replacement. Thus,

```
\snfig{Caption title.}{file-name}{.5}
```

is turned into

```
\snfig{Caption title.}{file-name}{.5}{}
```

and the same is done for every other instance of `\snfig` in each case properly maintaining the argument text through the replacement step.

The above example may need some tweaking if the instances of the command being changed sometimes span new line boundaries, but typically this also can be handled, as can be much more complex instances of detecting what should be replaced and what should not be in various instances. In fact, depending on the editor’s particular regular expression capability, the earlier example of replacing en-dashes by hyphens perhaps could have been done with one Replace All using a regular expression to search of `-- not` followed by a third `-`.

In section two I recommended that anyone not already using  $\LaTeX$  macros should learn to use them. I recommend the same thing about using regular expressions if your editor supports them. They won’t be needed as often as macros, but when they are needed they are a major productivity increaser.

### 3.2 Other editor features

All of the serious text editors I have used allowed me to mark the cursor position with a couple of key strokes (e.g., Alt-F11 in WinEdt), move the cursor somewhere else (for instance to select some distant text and cut it, and then jump back to the first cursor position (e.g., Cntl-F11), where I might paste the text cut from elsewhere in the document.<sup>6</sup>

Text editors such as I have in mind also typically have provision to have multiple text buffers rather than just one cut-and-paste clipboard.<sup>7</sup> I gave an example of this in the regular expression example in subsection 3.1, where three bits of text were simultaneously saved from the replaced string of characters for placement in the replacement string. With multiple places to save text, it is also possible, for example, to search-for-and-cut one

<sup>6</sup> I don’t claim the none of these features are available in various editors that are packaged with commercial versions of  $\TeX$ ; I hope they are. I am only suggesting that you find an editor that supports such capabilities. What I do know is that with each new release of MS Word I find it harder and harder to find such features, if they exist at all, while they are easy to find in the two text editors I currently use regularly (WinEdt and EMACS).

<sup>7</sup> Alex Simonic, the developer of WinEdt, the editor I mostly use, showed me how to write macros to provide multiple text buffers in WinEdt.

bit of text, search-for-and-cut another bit of text, search-for-and-cut a third bit of text, and then paste together at some other point in the file, saving (in this example) several moves of the cursor in comparison with an editor with a single clipboard.

To give another example of the usefulness of multiple text buffers, I often find something on a web page and want to copy something from the page as a quote in a document I am writing *and* copy the URL as the source of the quote. Without multiple text buffers this requires the following sequence: (1) select text to be copied, (2) copy to clipboard, (3) switch window to other document, (4) position cursor, paste contents of clipboard, (5) switch window for first document, (6) select URL text, (7) copy to clipboard, (8) switch window to other document, (9) position cursor, (10) paste contents of clipboard. With multiple text buffers it requires: (1) select text to be copied, (2) copy to buffer A, (3) select URL text, (4) copy to buffer B, (4) switch window to other document, (5) position cursor, (6) paste contents of buffer A, (7) reposition cursor, (8) paste contents of buffer B. The latter method it not necessarily less key strokes (the macros I have for WinEdt take 12 steps), but it is somehow easier for me not to switch windows and have to refind my place as often.

The fact that  $\LaTeX$  is not locked to a particular editor also means that each participant in a collaborative project can use the editor with which he or she is most familiar. (Collaboration is also made easier because there is much more compatibility from release to release of  $\TeX$  and  $\LaTeX$ , even with multiple providers, than there is from release to release with many non- $\TeX$  commercial products. For instance, MS Word seems to go out of its way to enforce inter-release incompatibility in a way apparently aimed at forcing all collaborators to all upgrade to the same release.)

Using a text editor in conjunction with  $\LaTeX$  with its explicit markup also has advantages. For instance, it is easy to search for an italicized version of a word (i.e., search for `\textit{word}`) as distinct from a non-italicized version of the same word. Similarly, it is possible to search for all headings of a certain level (for instance, all instances of `\subsection`); with a system with implicit markup (e.g., MS Word) one might have to search for the words of each subsection title.

In subsection 2.1 I showed the use of `\include` files. This works because  $\LaTeX$  has the provision for specifying in one file a list of files to be included as if they were text in that first file. The editor I mostly use, WinEdt, also supports this; it knows enough about  $\LaTeX$  to search the highest level file for instances of `\include` and gives me a list of visual tabs to the various files to be included; this makes it very easy for me to move among the various in a longer document.

I could give an unlimited number of examples what powerful text editors can do once one breaks free of the limitations of hidden, proprietary, undocumented markup and those built-in editors whose graphical user interfaces eliminate powerful editing capabilities as part

of providing a point-and-click environment to the user. One of the best things is that I can use the same editor with which I have become facile from application to application. (Also, if several authors are collaborating, they can all use their own preferred editor, and — perhaps more importantly — they don't all have to have the same version of Word installed.)

## 4 Conclusion

To conclude, I give some additional opinions on a couple of thoughts I hinted at in the earlier sections and one additional opinion.

### 4.1 Conservation of hassle

In my observations covering many decades, it has always taken many fussy steps to do anything involving typesetting for reproduction. The computer era has eliminated many physical steps, each of which required its own sort of skill and complexity. However, the computer era hasn't done anything to decrease the total number of steps — they are just done with a keyboard and mouse now and the skill is in knowing what commands can do what you need and where to find them. What computer-based approach is best is a matter of personal choice — they are all filled with hassle. My own choice has evolved to be a powerful, explicit typesetting language (L<sup>A</sup>T<sub>E</sub>X) combined with a powerful text editor. Some people argue that the WYSIWYG approach is easier. On average, however, I see the WYSIWYG approach as just as much work for the same task: A simple task in Word also tends to be simple in L<sup>A</sup>T<sub>E</sub>X; a more advanced task (e.g., inserting a cross-reference) seems more straightforward to do in L<sup>A</sup>T<sub>E</sub>X than in Word. As one uses more and more of L<sup>A</sup>T<sub>E</sub>X's power, the same task typically seems more and more difficult to do in Word as well, and my frustration level seems to grow faster with Word. Since most users use Word in only trivial ways, Word is pretty trivial to use at that level, but then so is L<sup>A</sup>T<sub>E</sub>X at that level.

### 4.2 The assertion that L<sup>A</sup>T<sub>E</sub>X is hard to learn

I have no doubt that most people could learn to use L<sup>A</sup>T<sub>E</sub>X and a good text editor if they saw it as beneficial; L<sup>A</sup>T<sub>E</sub>X and a text editor at the intermediate level of sophistication at which I use them are no more complex than trying to use Word for the same task (and I mostly think they are less complex than Word). Think about all the other, fairly complex things people master in their lives — cooking, knitting, growing flowers, fly tying, and the rules of baseball. By comparison, there is nothing inherently too difficult about using L<sup>A</sup>T<sub>E</sub>X — it's only a question of learning enough to see its comparative benefits (and cheaper price). And anyone who can learn to use Word at a high level with all its particular weirdnesses (and changes to the user interface with each release) can surely also learn to use L<sup>A</sup>T<sub>E</sub>X at the same level.

### 4.3 Using what everyone else is using

I believe the main argument against L<sup>A</sup>T<sub>E</sub>X and for Word is ubiquity of use. People use Word because everyone else does — their collaborators, their publishers, etc. — not because Word is better. If people were interested in a better word processor, they would use Word Perfect or perhaps one specialized to their area of writing such as Note Bene. If the world of T<sub>E</sub>X is to have the best chance of snagging and keeping potential L<sup>A</sup>T<sub>E</sub>X users, we must offer them a great, if small, community of fellow users. This brings me to my “pet peeve” about our T<sub>E</sub>X user groups. I suggest that no question, however basic, should every be answered with “read the manual.” If reading the manual is what should have been done, the answer should be, “I think page N of document Y at URL Z” tells you what you want to know. In the real world (i.e., the world of MS Word) people don't read the manual (recognizing this, Microsoft for the most part has stopped providing documentation, and what documentation they do provide is hard to access and understand). If I have a question about Word, I ask the person at the next desk, for example, “How do I move the margins in for a block quote?”; and the person I ask either shows me without comment or first says, “Oh, that's easy [perhaps thinking what an idiot I am for not even knowing that]” and then shows me how. More generally, most people don't read manuals about most things — they just bash along learning by trial and error and by asking people how to do things. I have two challenges for people who monitor the T<sub>E</sub>X user discussion groups:

1. For people who are tempted to say “read the manual”: Try to say nothing at all. It makes us less knowledgeable users feel pretty bad and chary of asking another question of the group if there is any hint from the answerer that we are as dumb or out of place as we already secretly feel we are.
2. For intermediate and more experienced beginning users: Don't hesitate to answer when you know a useful answer even though you may be pretty much of a beginner yourself. If we who can still remember how little we felt we knew when we first started answer quickly, our intervention can perhaps preclude another less experienced user getting one of the those dreaded read-the-manual answers. Also, we may learn something more ourselves when someone more knowledgeable follows our suggestion with an improvement.

### Acknowledgements

I owe thanks to many sources for what I have learned about using L<sup>A</sup>T<sub>E</sub>X — books, the `comp.text.tex` list, the `texhax` list, and many individuals. Of course, none of them are responsible for lessons I have mislearned.

I can't remember and acknowledge everyone who directed me to techniques illustrated in this column; however I can remember some of them. Karl Berry reviewed an early version of this paper and earlier told me about

David Walden

some of the methods I have described here. I also remember Peter Flynn, Steve Peter, and Steve Schwartz telling me about particular techniques. Peter Flom, Will Robertson, and anonymous reviewers provided many helpful suggestions for section 2 which appeared in an earlier incarnation in the 20006-2 issue of *The PracTEX Journal*. Will Robertson also carefully reviewed the complete paper here and made many substantive suggestions for improvement as well as catching many minor errors.

### **Biographical note**

David Walden is retired after a career as an engineer, engineering manager, and general manager involved with research and development of computer and other high tech systems. More history is at [www.walden-family.com/dave](http://www.walden-family.com/dave).

TYPESETTING AND EDITORIAL SERVICES  
ON-SITE TRAINING

Steve Peter  
310 Hana Road  
Edison NJ 08817

I specialize in foreign language, linguistic, and technical typesetting, providing a complete workflow, from rough manuscript through camera-ready copy. Book and typeface design is also available.

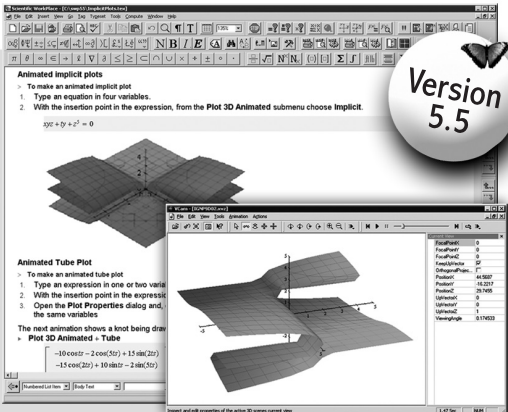
I am an award-winning teacher, and I offer training in all matters  $\text{T}_{\text{E}}\text{X}$ , as well as using text editors to their fullest, scripting, and basic programming. Training is targeted to the specific needs of the client.

Phone: +1 732 287-5392  
Email: speter@dandy.net

# Scientific WorkPlace<sup>®</sup>

# Scientific Word<sup>®</sup>

Mathematical Word Processing  
 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  Typesetting  
Computer Algebra



**New in Version 5.5**  
*Scientific WorkPlace and Scientific Word*

- ◆ Import  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  files produced by other programs
- ◆ Use many new  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  packages
- ◆ Select and right click a word for links to internet search options
- ◆ Get help from the new, extensive troubleshooting section of *Typesetting Documents with Scientific WorkPlace and Scientific Word, Third Edition*

**Scientific WorkPlace**

- ◆ Compute and plot using the *MuPAD*<sup>®</sup> 3 computer algebra engine
- ◆ Animate 2D and 3D plots using *MuPAD*'s *VCAM*
- ◆ Rotate, move, zoom in and out, and fly through 3D plots with new OpenGL<sup>®</sup> 3D graphics
- ◆ Label 2D and 3D plots so that the label moves when you rotate or zoom a plot

**The Gold Standard for Mathematical Publishing**  
*Scientific WorkPlace* and *Scientific Word* make writing, sharing, and doing mathematics easier. A click of a button allows you to typeset your documents in  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ . *Scientific WorkPlace* and *Scientific Word* enable both professionals and support staff to produce stunning books and articles.

**MacKichan**  
SOFTWARE, INC.

www.mackichan.com/tug • Email: info@mackichan.com  
Toll-free: 877-724-9673 • Fax: 360-394-6039

---

**Do you need on-site training for L<sup>A</sup>T<sub>E</sub>X?**

*Contact Cheryl Ponchin at*

`cponchin@comcast.net`

Training will be customized for your company needs.

Any level, from Beginning to Advanced.

---

XX

**Carleton Production Centre**

HUMANITIES TYPESETTING

Specialising in Linguistics

Since 1991

---

613-823-3630 • 15 Wiltshire Circle

Nepean, Ont., Canada • K2J 4K9

XX

NO MATTER IF  
YOUR DOCUMENTS  
ARE

BIZARRE  
smokin'  
friendly  
FORMAL

OR JUST A LITTLE

$$\frac{\text{odd}}{2} \neq \left\lfloor \frac{\text{odd}}{2} \right\rfloor$$

THEY'RE SPECIAL  
TO YOU

SO BRING THEM  
TO US

**silmaril**  
consultants

- ▷  $\text{\LaTeX}$  typesetting
- ▷ XML document systems

<http://www.silmaril.ie/>

O'REILLY

**OSCON**  
OPEN SOURCE  
CONVENTION

**JULY**  
**24 – 28**  
**2006**

**PORTLAND**  
**OREGON**



**OPENING**  
**INNOVATION**