The PracTEX Journal, 2010, No. 1
Article revision 2010/6/7

Travels in TEX Land: *memoir*, T$_T$H, and a booklet signature

David Walden

Abstract In this column in each issue I have mused on my wanderings around the TEX world. In this issue I describe three efforts. In section 1, I describe my first attempt to use the *memoir* class to produce a book. In section 2 (page 6), I describe my first time using T$_T$H to convert from LATEX to HTML. In section 3 (page 9), I describe creating a 16-page booklet signature using a method described by another author in an earlier issue of this journal.

This will be my final TEX Land column in this journal. I am pleased to have provided a column for every previous issue, but it is now time for me to focus on other things. I won't stop using TEX, however, and probably will continue to write about TEX once in a while, but without the concerns of a regular column. I wish the editors of this journal "all the best" as they continue publication of *The PracTEX Journal.*

1 Using *memoir*

Once before I briefly tried the *memoir* class for something small — I can't remember what. As I began to draft a recent book, I decided it was time to try *memoir* for a substantial project.

Starting out

First I downloaded a copy of the manual from

www.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf

and saved it in the directory of my book project for easy access, and I put the following command at the beginning of the include file for my book:

```
\documentclass[book,b5paper,showtrims]{memoir}
  %b5 = 176 x 250 mm = 6.8 x 9.8 inches
```

Copyright © 2010 David Walden

Note that I used *memoir's* `showtrims` option to the `\documentclass` command as I wanted to be able to see how the typeset text fit on the actual book size page. (I will drop this option just before sending the file to the printing company because I have no experience with having trim marks in a file going to a printer and printers have successfully trimmed pages from book files I sent without trim marks.)

I did nothing else except use *memoir* just as if I was using LaTeX as I drafted the books for several months.

Some time later, I added another command not available in LaTeX

```
\tightlists  %close up spacing for itemize, enumerate, etc.
```

and I went several more weeks without thinking about *memoir* again.

Finally, as I neared completion of the first draft of the book, I decided to change the text block size within the page size. I printed out a copy of chapter 6 of the *memoir* manual, glanced over it, and tried different commands that I thought should change the text black size or the margin sizes. Nothing worked. Eventually, I sought help from `comp.text.tex` and found a January 4, 2002, exchange of messages where *memoir* creator Peter Wilson reminded someone about the necessity of issuing the

```
\checkandfixthelayout
```

before the

```
\begin{document}
```

command. I did this, and then the margin and text block size commands had more reasonable effect.

However, I still could not (immediately) get the exact layout I wanted. The *memoir* system was checking for correct arithmetic or correct styling (I don't know which) and giving error messages. Since I needed to finish my first draft, I decided to not worry about the text block size and position any more at that time; I was able to achieve something closer to what I wanted, and there would be plenty of time to struggle with further refinement later.

So far, using *memoir* had not *required* much new study (although undoubtedly I would have benefitted from doing a little more study rather than doing almost none).

Bibliography, footnotes, and references

A few weeks went by after I drafted the previous section. I had continued drafting my book and struggling with a problem that I have had with previous books and papers — how to handle footnotes, the bibliography, and references in the main text to items in the bibliography.

In my three previously published books, I have in each case bowed to the marketing wisdom or publisher's demand that footnotes not be at the bottom of pages and instead be endnotes. In the first published book, the notes are at the end of each chapter, referenced by superscript numbers in the main text, and bibliographic entries are mingled with the other notes. In my second published book, there are again notes at the end of each chapter referenced by superscript number in the main text; however, all the bibliography entries are at the back of the book in alphabetical order by first author's last name and referenced by sequence numbers in square brackets in the main text. In this case, the bibliographic entries were done with `bibtex`. In my third (self-published) book, all of the notes are at the end of the book in separate sections for each chapter with the notes referenced from the main text by superscript numbers which restart at 1 for each chapter. In this case, I created the bibliography manually (not using `bibtex`) in alphabetical order by first author's last name and used (unconventionally) the first author's last name and publication year in square brackets in the main text, e.g., [Walden98] or [Walden05b]; I like this unconventional convention quite a lot and would have used it again in my current book if I knew how to make `bibtex` do the work for me.

My decision with the current book was to put all of the notes and bibliographic entries in a single References list at the end of the book, referencing the list with superscript numbers in the main text. I looked in the *memoir* manual for how I could do this, but didn't find anything relevant and then went to `comp.text.tex`, where I found the following information from Boris Veytsman in July 1997:[1]

1. Yuri Robbers says, "I think the natbib package (an add-on for bibtex) is able to provide bibliographic references according to your unconventional convention. If not, the makebst package will definitely be able to create a `.bst` file according to that convention, and it only requires you to answer a load of questions in order to do its job."

   Yuri is undoubtedly correct. However, the approach I used didn't take much time to implement, and I suspect I was happier going in this ad hoc way than bothering to learn a new package.

> ... to make footnotes appear as end-notes intermixed with citations is rather easy. Just create a file footnotes.bib with all your footnotes written as @misc entries like this:
>
> ```
> @Misc{footone,
>    note ={This is a footnote}}
> ```
>
> Then in the body of your document instead of the command
>
> `\footnote{This is a footnote}`
>
> put
>
> `\cite{footone}`
>
> In the list of your bibliography files put `footnotes.bib` like this:
>
> `\bibliography{physics,math,footnotes}`
>
> and you are done.
>
> If you need the labels to be superscripted instead of normal square brackets like [1], install the cite suite from CTAN and put
>
> `\usepackage{cite}`
>
> in the preamble of your document.

So I did that and it all worked except that the numbers on the items in the References list at the end of the book were still in square brackets. I looked in the `cite.sty` file and found that I needed another cite option, i.e.,

`\usepackage[super,biblabel]{cite}`.

That worked very well, until I decided to make the text of the References text be `\small`, i.e.,

```
\backmatter
\renewcommand{\bibname}{References}
\small
\raggedright
\bibliography{biblio}
```

at which point the superscripted numbers on the References list entries became almost too small to read. I read a little more of `cite.sty` and decided I needed

```
\usepackage[super]{cite}
\makeatletter
\renewcommand\@biblabel[1]{#1.}
\makeatother
```

which put unsuperscripted numbers the same size as the rest of the text in front of each entry in the References list.

Another issue had to do with URLs which were alone in the reference list. If in the main text I reference an organization's website, I often wanted to include the URL in the References list, e.g., "...CII's website`\cite{CIIurl}`" Then, in the References list I put an entry such as

```
@misc{CIIurl,
note="\url{www.ciionline.org}"
}
```

However, that had the problem that `bibtex` added a final period after the URL, and could confuse a reader. Once again I queried `comp.text.tex` and Lars Madsen responded with the following trick:

```
\def\myurl#1#2{\url{#1}}
...
@misc{CIIurl,
note="\myurl{www.ciionline.org}"
}
```

where `\myurl` uses TEX's macro calling process to throw away the extra period inserted by `bibtex`. (Lest Lars be faulted, he told me to use `\newcommand`, i.e.,

```
\newcommand\myurl[2]{\url{#1}}
```

rather than `\def`. I don't have any valid excuse for not using `\newcommand`.)

Finishing

Another couple of weeks later, I once again felt I was nearing completion of drafting the book and it seemed like time to think about part and chapter running

headings, which I have used in my previous books done in LaTeX and not *memoir*. This seemed pretty complicated in LaTeX. From reading the manual, creating running headings seemed much simpler in *memoir*. However, after thinking about it for a little while, I decided to skip having running headings for this book, for a lighter if less informative look.

I guess I am going to seek reviews with the formatting I have now, which I am thinking will be my final formatting. I haven't really used many of *memoir's* capabilities, but the few that I have used have been easily accessible. It has been a great convenience to have the *memoir* manual readily available in the directory of my book files, and I have often gone to it.

In the previous section on "bibliography, footnotes, and references," I described how I handled the bibliography, footnotes, and references in this book. It occurs to me now that I didn't bother to look in the *memoir* manual to see if it already had a mechanism for what I wanted to do. In any case, I'm not going to make any changes in that handling at this late date.

All in all, I made the change from using LaTeX to using *memoir* with minimal effort and with less need to explicitly load additional packages. I assume that if I had used more of *memoir's* capabilities, I would have found it even more useful.

Acknowledgments for this section

Yuri Robbers reviewed this section, made several suggestions for improvement, and caught a number of typos. Thank you, Yuri.

As always, the people past and present at `comp.text.tex` provide much support. I also enjoy the fact that MiKTeX automatically loads packages when one gives a `\usepackage` command for a package that is not already installed. Finally, I benefit every day from the packages, format, classes, etc., that members of the TeX community write or have written; I don't think often enough to thank them.

2 Trying T$_T$H to solve a problem

Publishers all too often want a submission to arrive as an MS Word file which a copy editor will then edit. Next in my experience, the publisher flows the content of the Word file into a typesetting program such as InDesign; or the publisher

may actually format the Word file for publication and generate a PDF file from that.

More than a decade ago I decided to stop using Word for composing documents submitted for publication, and LaTeX (along with MiKTeX) has been my system of choice for composing and formatting documents I am going to later submit for publication. However, once it is time actually to provide the document to the publisher, I am faced with converting it into Word, because that is what the publisher wants.

I have used various ad hoc methods for going from LaTeX to Word. For instance, for an 800-page book I used VisualTeX to generate HTML directly from the LaTeX for each chapter and then loaded the HTML into a Word file for each chapter. For small documents, I often have copied-and-pasted the text of my LaTeX document into Word and then manually in Word removed all the LaTeX commands and executed the Word formatting commands. For a while, I used TeX2Word (http://tug.org/pracjourn/2005-4/walden/), until I changed computers and scrapped the computer on which the program was installed.

A few months ago I again had to convert a LaTeX document into Word. I had sent the final draft to the editor of the journal as a PDF file. However, the journal's publication approach was to use a Word template to format documents for publication on the web (it is an electronic-only journal) and then output the formatted Word document as a PDF with a link from an HTML cover page in the on-line journal. I decided it was time to try one of the TeX-to-HTML conversion programs and to again get to Word's `.doc` format via HTML.

I googled for "`LaTeX to Word conversion`" and found myself at a webpage on the TUG server: http://www.tug.org/utilities/texconv/textopc.html entitled "Converters from LaTeX to PC Textprocessors — Overview." I skimmed to the section called 'HTML as intermediate format," and following the link to the homepage for `TeX4ht`. However, there the first paragraph under "Installation" said, "To be installed, the system needs a port made up of native utilities of TeX4ht and of non-native utilities. The easiest way to establish an up to date port is to download an installed distribution of the system, and upgrade it with the files provided here." That didn't sound as easy as I like installations to be (I hate configuring computer programs).

The next program listed on the TUG server page after `TeX4ht` was T_TH. So, I followed the link to its home page (http://hutchinson.belmont.ma.us/tth/)

and continued following the links to "T$_T$H Distribution," "link to download list," and "windows executable." Clicking on the last of these caused a `.zip` file to be saved on my computer. I unzipped it, opened the file `tth_manual.html`, read a few lines under "Usage," and tried it under Cygwin (I run Windows XP):

```
tth.exe <myfile.tex
```

I struggled for a few minutes with the fact that `tth.exe` was not being found by Windows XP (downloading T$_T$H didn't install it and update the execution search path). But eventually I conquered that problem, and the above command tried to compile my LaTeX file into HTML.[2]

However, T$_T$H complained about some of my LaTeX commands. It didn't recognize `\thinspace` and it didn't know about the `paralist` and `geometry` packages. I commented out the commands to use the *geometry* and *paralist* packages and changed the `paralist` commands to standard list commands, e.g., from `\begin{compactitem}` to `\begin{itemize}`. I also deleted my use of `\thinspace`. The publisher would reformat all this stuff anyway.

I can't remember if there were a couple of other LaTeX commands T$_T$H didn't recognize, or not. In any case, after a few such changes, T$_T$H compiled my whole LaTeX file into HTML.

However, instances of `---` failed to produce the correct thing in HTML. Rather than bothering to figure out why, I replaced the three instances of `---` with the code `*MDASH*`, which passed through the HTML phase and into Word where I did a Replace All of `*MDASH*` with an em-dash.

My writings never include significant math, so I don't know how T$_T$H would handle that. But for my non-math document, the HTML looked pretty good. I started Word, opened the HTML file from within Word, and it still looked pretty good — good enough for submission to the publisher's editing and typesetting process which would be reformatting the document according to their style. I did do a little editing of paragraph indents in the the Word file in addition to the `*MDASH*` replacements mentioned in the previous paragraph. I finished the process by saving my document as a `.doc` file from Word.

Using T$_T$H will be at the top of my list of options next time I need to go from LaTeX to HTML or Word, even if a little final editing in HTML or Word is

2. Probably the commercial version of T$_T$H (http://hutchinson.belmont.ma.us/tth/tthgold.html) avoids some or all of the problems mentioned in this and the next paragraph.

required.

For comparison, my LaTeX, PDF-from-LaTeX, and HTML-via-TtH, and Word-by-opening-HTML files are available via links on the HTML page for this note.

3 Creating a booklet signature

Recently I wanted to create a 16-page booklet just by folding in half 8-1/2-by-11-inch pages of paper that are normal in the United States. I remembered an article by D.V.L.K.D.P. Venugopal entitled "Creating Pocket-size Books Using LaTeX" that appeared in an issue of *The PracTeX Journal* (tug.org/pracjourn). Using the author index of the journal, I found the article in issue 2006-3 (http://www.tug.org/pracjourn/2006-3/venugopal-pocketbook).

To create my 16-page booklet, I closely followed Venugopal's first couple of steps, except using a different size sheet of paper as the starting point.

I first created a LaTeX file to produce a file called `content.pdf` containing the content of the booklet on sixteen 5-1/2-by-8-1/2-inch pages. The LaTeX file `content.tex` had the following structure:

```
\documentclass{article}
\usepackage{geometry}
\geometry{papersize={5.5in,8.5in},lmargin=.5in,%
        rmargin=.5in,tmargin=.5in,bmargin=.5in}
\begin{document}
... ... ...
\end{document}
```

Then, still following Venugopal's model, I created another LaTeX file called `signatures.tex` as follows:

```
\documentclass{article}
\usepackage[final]{pdfpages}
\begin{document}
\includepdf[pages=-,nup=1x2,landscape,signature=16]{content.pdf}
\end{document}
```

I compiled the file `signatures.tex` and then printed out the file `signatures.pdf` using my laser printer's option for printing on both sides of a sheet of paper. The pages were printed such that when I folded the resultant four sheets of paper in half, I had the desired 16-page booklet with the pages in the proper order and orientation.

Once again, an article in a back issue of *The PracTEX Journal* allowed me to copy what someone had done before rather than learning something on my own.

Acknowledgment

Barbara Beeton caught almost a dozen typos in the abstract and three sections of a prepublication draft of this column.

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. He holds an undergraduate math degree and completed a graduate school sequence of courses in computer science. More history is at www.walden-family.com/dave.