The PracTEX Journal, 2008, No. 2
Article revision 2008/6/20

Travels in TEX Land: A bigger experiment with ConTEXt

David Walden

Abstract In this column in each issue I muse on my wanderings around the TEX world. In my column in the 2007-2 issue (http://www.tug.org/pracjourn/2007-2/walden/) I tried a small experiment with using ConTEXt. In this issue I describe an additional, quite extensive, effort to use ConTEXt — to create a picture book for a "slide show" I was involved in creating a number of years ago.

Background regarding the book

I live on the edge of a salt marsh on Cape Cod in Massachusetts. In 1991 my son, a photographer, and I undertook a two week summer vacation project to study the marsh. He took the 35mm photos and I was the photographers assistant. We both did research about the marsh in the library of our local nature center. We wrote a script and recorded it on cassette tape. And on the second Saturday of our vacation, after sunset when it was dark enough for the slides to show great colors, we presented our slide show of the marsh to my wife, manually changing the slides to follow the tape recorded narrative.

In the following years, I presented the slide show on several occasions to small groups of guests in our home. Eventually I got the idea that I should create a CD version of the slide show that I could give to people to watch on their own computers.

In about 2000 I paid Kodak to scan the 212 slides in our slide show and to give me a CD with high resolution digital versions of the slides. In 2005 I finally got around to creating a CD of the slide show, using basic HTML.

First I processed every digital slide with Photoshop to create cropped, color-adjusted, low resolution .jpg files of each slide suitable for computer screen display without taking up as much space as the scans from Kodak did. Next, I edited the Word file of the script of the slide show to a a driver file which could be processed with a Perl program; see `scorton-marsh-script.txt` on the HTML page for this column. Notice the code (p1 through p6) in the third argument of each entry in the drive file which specifies how to format different styles of HTML pages depending on the orientation of the image, pages with section titles, and so forth.

The Perl program generated an `.html` file with navigation links; you can see an example of a few "pages" of such output at `marsh-slideshow/001.htm`. I copied a directory containing `.jpg` and .html files onto a CD, printed the CD art on a circular adhesive label, pasted the label on the CD, and gave people the CD with instructions of how to click on the first `.html` page in sequence on the CD.

Recently I have been experimenting with self-publishing (as described in several of my columns), and I decided that it would be nice to create and publish a picture book from the slide show. This is the subject of the rest of this column.

Initial thoughts on the book

My first mental image of the picture book had a marsh photo on each page at a specific spot relative to the top and side margins with the text for the page in a separate place on the page. For this it seemed to me that I needed to avoid using most of TEX's float capability and ideally would have several templates that would determine the layout of each of several types of pages. I guessed this would be easier to do in ConTEXt than LATEX, although this was not a well-informed guess. In any case, I decided to produce the book using ConTEXt — it couldn't hurt to get some significant experience with ConTeXt.

I asked Aditya Mahajan (who is writing a series of articles on ConTEXt for *TUGboat*) whether ConTEXt had the template capability that I envisioned, and he directed me to ConTEXt's layers capability: `http://wiki.contextgarden.net/Layers`. With that assurance that I could do what I wanted with ConTEX, I began to create a ConTEXt file that had a book format. The first, sketchy effort, is documented in the files `book-initial.tex` and `book-initial.pdf`.

Creating the basic ConTEXt program

At this point, it seemed like it would be useful to generate a ConTEXt program for the entire book which generated appropriate macro calls for each type of book page based on my existing `scorton-marsh-script.txt` file. In other words, it was time to modify my Perl program, which seemed a relatively easy prospect since all it had to do was generate a macro call for each type of output page and I could define the macro calls later in ConTEXt (and, thus, modifying the program seemed mostly to involve skipping over code to output various lines of HTML).

In fact, it took me about an hour to modify the program; see `slides2tex.pl` if you are interested. With an an almost TEX program in hand, I manually modified the file to remove angle brackets, to at least initially try to capture section titles embedded in type two slides, etc.; this took another hour to get a TEX file that PDFTEX could compile: `scorton-marsh-script.tex`. I figured that if PDFTEX could compile the file, then I was not too far from something ConTEXt could compile.

If you look at that TEX file, you will see that each type of slide went to the same macro definition. Now it was time to define separate macros for each slide type — and to figure out some actual lines of ConTEX for these macros. I merged the contents of texttscorton-marsh-script.tex into my `book-initial.tex` ConTEXt file, called the merged version `book-in-context.tex`, and made a few necessary trivial modifications to the script part.

Within another hour or two I had compiled the draft book in ConTEX; see `book-in-context.tex` and `book-in-context.pdf` for an example of this version before detailed formatting was begun.

The power of the command-based approach to typesetting of TEX is never more apparently than when TEX is used for a project like this which involves converting content from a markup language in an existing file and the pages of the document can be described with only a few templates.

By the way, my approach to learning what I needed to know about ConTEXt to get this far was to go to `contextgarden.net` and search for words involved with whatever I needed to know or to search Hans Hagen's *ConTEXt Manual* at `http://www.pragma-ade.com/show-man-9.htm` or other manuals at `http://www.pragma-ade.com/show-man-1.htm`.

Refining the ConTEXt program

Next I needed to deal with the detailed formatting in ConTEXt and to return to my high resolution scans from Kodak and reprocessing them to get print quality images. Such reprocessing of the images to get appropriate cropping and good looking colors took me a several hours a day over two or three weeks the last time I did it, so I started with the formatting issues.

I wanted to have a little space between paragraphs of text on the photo pages. Using the `\setupwhitespace` put a space between paragraphs but I was unable to control the height of the space with the argument of the command; the space was always the same size for the parameter values I selected. Then I discovered that I had to turn off typesetting on a grid, which was in a command left over from the file for my earlier column which was my starting point for this project.

I noticed that ConTEXt was numbering pages starting with 1 for the cover page of the book and, thus, the first page after the front matter was numbered 4. The front matter pages didn't have page numbers which was OK—I had no need to give them page numbers using small Roman numberals. I searched `contextgarden.net` and found the command `\setnumber[page]{1}` which I added after the front matter in the ConTEXt file.

I wanted to use PDFTEX's capability to do micro-typesetting, both protrusion beyond the end of the line and font expansion and contraction. However, I couldn't find anything about micro-typesetting by searching the `contextgarden.net` or the *ConTEXt Manual*. Eventually Aditya Mahajan directed me to `http://wiki.contextgarden.net/Protrusion` which gives various synonyms for micro-typesetting but doesn't use that term exactly. Eventually I found a combination of commands that effected micro-typesetting:

```
%\usetypescript[serif][hz][quality]
%\setupalign[hanging,hz]
\usetypescript[serif,sans,mono][hanging][pure]
\setupalign[hanging]
\usetypescript[modern-base][texnansi]
\setupbodyfont[reset]
\setupbodyfont[modern]
```

These commands did slightly different micro-typesetting on a few page test case with the first two lines above commented in and the next two lines commented out. However, when I tried either sequence of commands with the entire book, pdfTEX came to a fatal error saying it had run out of space and something more about a hash table. With a little testing I discovered that commenting out the last three lines above let the whole book compile, but then I didn't appear to get micro-typesetting. When I asked the ConTEXt mailing list (ntg-context@ntg.nl) about this, Hans Hagen suggested that I "increase the hash size — this is done in texmf.cnf; after doing that you need to remake the formats." However, I could not find texmf.cnf on my computer (I am using whatever version of ConTEXt came with the version of ProTEXt from a couple of years ago), and I didn't know what "remake the formats" meant. I have little interest ever in struggling with configuration issues, and definitely didn't want to hear that I needed to get a newer version of ConTEXt, so I gave up on using micro-typesetting; instead I changed the text wording in the places in the sparse text of this picture book to eliminate the need for micro-typesetting.

By the way, I do all of my ConTEX compiling using texexec. I also always wish that ConTEXt had a different name so that Googling on it and some other keywords didn't find all of the instances of the keywords in whatever "context" they appear.

My next question was how to get an even number of pages in the front matter such that page 1 of the main matter started on an odd numbered page. Another search of contextgarden.net (for "double-sided") took me to a discussion of FO (whatever that is) page layout including the command

```
\setuppagenumbering[alternative=doublesided, state=stop]
```

I didn't know what the state=stop part of the command means, so I deleted it and the remaining command seemed to do what I wanted it to do. It also started each new chapter on an odd numbered page, and considered the possibility of turning off double-sided page numbering for the main body of the book — a final decision which could be made later.

Of course, every change can bring up new issues, for example, with double-side page number, how could I leave an even numbered page at the end of a chapter completely blank — no page number. After searching contextgarden.net and the manuals at www.pragma-ade.com for an hour or so without figuring out how to do this, I sent a message to the ntg-context@ntg.nl mailing list. I immediately received a message back from Jörg Hagmann indicating I should learn about learn about \definepagebreak. I searched contextgarden.net and the archives of ntg-context@ntg.nl (http://www.ntg.nl/pipermail/ntg-context/) for \definepagebreak, found a long discussion about what worked and didn't work for various people, but I couldn't make any of those ideas work and eventually gave up. Instead I simply changed my macro for the start of chapters so it does a \setuppagenumbering[state=stop] before the \chapter call to turn off page numbering for any blank page generated at the end of the previous chapter and does a \setuppagenumbering[state=start] after some text for the chapter has been generated which turns page numbering back on for the first page of the chapter (which is how I want it to be).

However, despite how I left things at the end of the last paragraph, Aditya Mahajan sent another response to the ConTEXt mailing list asking me if the following commands (which I have paraphrased to match my actual situation) had the right result.

```
\definepagebreak
  [mychapterpagebreak]
  [yes,header,footer,right]

\setuphead
  [chapter]
  [page=mychapterpagebreak]

\setuppagenumbering[alternative=doublesided,location=footer]
\setupfootertexts[{\ssx {\pagenumber}}]

\starttext
  \chapter {testA} \dorecurse{10}{\input tufte }
  \chapter {testB} \dorecurse{10}{\input tufte }
  \chapter {testC} \dorecurse{10}{\input tufte }
\stoptext
```

I tried the above test case and it worked, so I adapted it to my actual book file and removed the start- and stop-numbering commands I described at the end of the last paragraph.

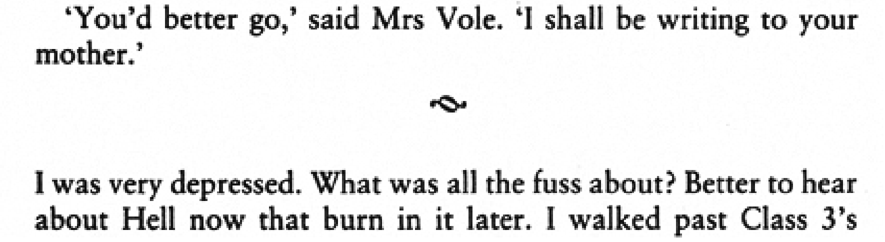
I cannot avoid any longer the giant job of reprocessing my original scans for Kodak. Thus, I'll stop working with ConTEXt for now. The current source file for my book is `book.tex`, and an example of a few pages from the typeset book is `book-example.pdf`. I'll try to remember to update this column with information on how to buy the printed book when it is available for sale.

Follow-up on my prior column

In my column in TPJ issue 2008-1, I once again discussed what I call "thought breaks," which I previously had discussed in issue 2005-4.

I am becoming more and more sensitive to these ornaments used for typesetting such thought breaks. A few weeks ago I came upon another ornament that I particularly like — in Jeanette Winterson's book *Oranges Are Not the Only Fruit*, Grove Press, New York, 1987. This book uses a variation of the same symbol for smaller and bigger thought breaks, as follows.

Smaller break example:

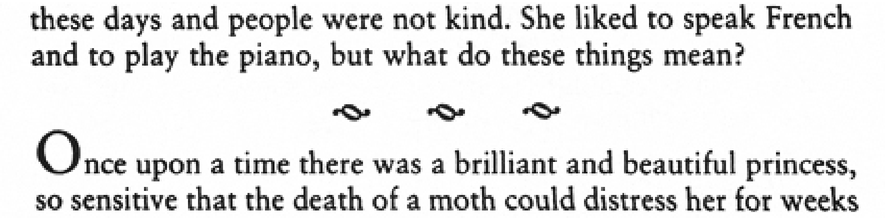


'You'd better go,' said Mrs Vole. 'I shall be writing to your mother.'

⁓

I was very depressed. What was all the fuss about? Better to hear about Hell now that burn in it later. I walked past Class 3's

Bigger break example:



these days and people were not kind. She liked to speak French and to play the piano, but what do these things mean?

⁓ ⁓ ⁓

Once upon a time there was a brilliant and beautiful princess, so sensitive that the death of a moth could distress her for weeks

Acknowledgments

Aditya Mahajan answered key questions about using ConTeXt for this project and provided other guidance. Responses from participants in the ConTeXt mailing list were also helpful. Yuri Robbers made several suggestions for improving the final manuscript.

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.