"USING VFP TO CREATE LaTeX DOCUMENTS"

Presentation to the
Atlanta Foxpro Users Group

July 20, 2006

5090 Hampton's Club Dr. ◆ Alpharetta, GA 30004 ◆ 678-528-3056
◆ kcully@cullytechnologies.com ◆ www.CullyTechnologies.com

ABOUT ME:

Kevin Cully is owner of **CULLY Technologies, LLC** that designs websites, creates database applications, conducts training and presentations. Cully Technologies, LLC has been in existence since 2000, and since that time, he has concentrated on building web sites, web applications, and fat client applications in a variety of commercial industries. Kevin has been programming in FoxPro since FoxPro 2.0 for DOS in 1992 and has been programming using the Web Connection framework since 1997.

CULLY Technologies is also a major sponsor of the Fox Forward 2006 conference to be held in Alpharetta, GA USA on September 15th, 16th, and 17th.

5090 Hampton's Club Dr. ◆ Alpharetta, GA 30004 ◆ 678-528-3056
◆ kcully@cullytechnologies.com ◆ www.CullyTechnologies.com

ABOUT TeX:[1]

TeX is a typesetting system created by Donald Knuth. Together with the METAFONT language for font description and the Computer Modern typeface, it was designed with two main goals in mind: first, to allow anybody to produce high-quality books using a reasonable amount of effort, and, second, to provide a system that would give the exact same results on all computers, now and in the future. It is free and is popular in academia, especially in the mathematics, physics, computer science, and engineering communities.

THE HISTORY OF TeX:

When the first volume of Knuth's The Art of Computer Programming was published in 1969, it was typeset using monotype, a technology from the 19th century which produced a "good classic style" appreciated by Knuth. When the second edition of the second volume was published, in 1976, the whole book had to be typeset again because the Monotype technology had been largely replaced by photographic techniques, and the original fonts were not available anymore.[2] However, when Knuth received the galley proofs of the new book on 30 March 1977, he found them awful.[3] Around that time, Knuth saw for the first time the output of a high-quality digital typesetting system, and became interested in digital typography. The disappointing galley proofs gave him the final motivation to solve the problem at hand once and for all by designing his own typesetting system. On 13 May 1977, he wrote a memo to himself describing the basic features of TeX. He planned he would finish it on his sabbatical in 1978, but as it happened the language was frozen only in 1989, more than ten years later.

1 About TeX and History of TeX from the Wikipeia article: http://en.wikipedia.org/wiki/Tex

MY BEGINNINGS WITH TeX:

I was beginning a new business venture where I wanted the final product to be a custom book. A book complete with table of contents, pictures, diagrams, chapters, sections, indexes, bibliographies. The whole deal. I was thinking that I could potentially create the book using MS Word, or OpenOffice.org's Writer. Perhaps I could even use the open sourced desktop publishing product called Scribus.

MS Word made me very nervous in that it seems that every new version of Word has some sort of impact on the automation model. In the worst case scenario, it would completely break the automation model. With Office 2007 looming, I didn't want to take that path.

OO.o Writer would be a good contender but I've never automated it before. Ted Roche has done some good articles on automating Writer and I could look into that but I'm talking of books with upwards of 200 to 300 pages. I'm not sure if it could really handle the generation of the index and TOC with this size of a book. I believe that Whil Hentzen uses OO.o products in the publishing of his books which is a good endorsement.

Scribus was also a contender but as far as I knew I didn't know of any articles on automating Scribus. I couldn't find any reference to generating TOC's and indexes either. I think it was more of a desktop publishing product, not specifically geared towards generating books.

In checking with my peers, several people recommended TeX. As an open source product, it not only is free, but runs on any platform, is specifically geared towards the publishing industry, and appears to do everything that I needed it to do. It was definitely worth looking into.

THE REALITY OF TeX:

As is typical with an open sourced project, there was documentation and examples of TeX (and LaTeX) but they were spread all over the web. Lots of examples, which is good, but it's hard picking through examples but not finding exactly what I'm looking for. There are newsgroups and forums, but I am so green, that I feel like I haven't passed the "RTFM" stage to really get in there to ask questions.

There is a good repository of utilities, packages, and fonts for the Windows operating system called MiKTeX. (http://www.miktex.org) This is a package manager for TeX similar to Synaptic is for Linux. These packages are essential for getting some of the more advanced features to function in TeX. We'll talk more about packages when we look into the VFP FoxToTeX class.

I knew that if TeX was going to be working out for my needs, that I shouldn't "hard code" the TeX markup into a specific program. I would be using TeX to generate several types of documents and not just my one book style. This is a case for a class if ever I saw one. The markup is simple, but spread through the web. This class is a way of documenting what worked for me in creating a TeX document. In some cased I was making the references more palatable for my own use; other times I was just formalizing a reference so that I could find it easily.

IN REGARDS TO TeX PACKAGES:

TeX (PDFLaTeX) is extremely fast in turning TeX files into PDF documents, but as you add packages, it slows the process down. In my VFP class, I have the methods self register a package as the methods need them by calling THIS.TeXAddPackage({my package name}). As an example, when you call the THIS.TeXIndexItem() method, it will self register the "makeidx" package which is needed in generating the index files. The "makeidx" package is not installed by default if you install TeX or LaTeX but if you use MiKTeX, it makes it easy to search and install the packages that you need.

TeX TABLES GIVE ME TROUBLE:

In my books, I'm wanting to turn some VFP into tables in the document. I figured that it would be a relatively simple task. I have literally spent days pouring over several ways of creating tables and I haven't satisfied myself with the results yet. If I wanted to create a HTML table in a web page, I could tell the table to take up 100% of the width of the display (screen on the web, or printed page from the printer), and the tables would self adjust their column widths if they weren't hard coded. TeX doesn't work this way. Even after trying several packages, I'm not really close to what I'd like to see.

This is where a Fox report really shines. If a table spans a page, I'd like it to reprint the header on the next page (and the next page after that), and to word wrap the columns where appropriate just like Fox does.

I've tried several approaches and several packages: table, longtable, tabular, tabularx, and some others. I'm getting closer with the longtable package but this will take more research. I've created another class inside of the FoxToTeX.prg just for this purpose. It's called "CursorToTable". My intention is that you can call it with a cursor and it will create a TeX table for you. More study needed.

SAMPLE VFP PROGRAMS:

Included are several programs that create TeX files and then call PDFLaTeX to create PDF documents out of them:

- Sample_Fox_To_Graphics.prg
- Sample_Fox_To_Math.prg
- Sample_Fox_To_PDF_Merge.prg
- If you call the FoxToTeX.prg file itself, it creates a document with tables that were created from VFP cursors.

JUST GETTING STARTED:

I'm certainly not a TeX expert. There is so much to learn about TeX and the METAFONT language that it would take years to learn the majority of it. TeX certainly is powerful and can be a great addition to the VFP report writer to help create polished report output. Hopefully by using my FoxToTeX class, it will help you add this into your arsenal of business solution tools.