

Re: Advice needed: Text editor with tags.

Source: <http://coding.derkeiler.com/Archive/Tcl/comp.lang.tcl/2004-03/0704.html>

From: Svenn Are Bjerkem (*spam_at_bjerkem.de*)

Date: 03/14/04

Date: Sun, 14 Mar 2004 22:37:40 +0100

Hi Neil,

I have read your posting with interest.

In article <nem-DE8DE1.19361610032004@larochelle-innovation.com>, Neil Madden <nem@cs.nott.ac.uk> wrote:

> *Well, on first reading, it sounds like you are after something like
> Docbook. With Docbook, theoretically, you are writing purely the
> content. You add tags to say what different parts of the text *are*, but
> you don't specify in the document how this should be presented in the
> output format. Now, in the world of XML (and Docbookx), you can use an*

It is, as you have understood, my intention to write text and care about what the text represent and not what it looks like. I don't even want to change font size to indicate a title or section header. Everything is courier 10 and all formatting is shown to the editor as different colors on foreground and background. As time pass by he will learn to remember the different colors. I would also implement a help color chart with proper keystrokes etc but that is way in the future.

> *When you want to create the output (such as PDF), you can run the XML
> through an xsl processor (such as FOP). The problem here is that most of
> the xsl processors don't seem too mature (well, of the free ones I've
> tried), and the resultant PDF is not great quality (especially when
> compared with the output of TeX). Writing an XSL stylesheet to produce
> your desired formatting is quite a mammoth undertaking, as well. I
> expect that this will improve as time goes on, but at the moment, I
> don't think it's quite there yet. You could, of course, write a
> converter to translate docbook -> latex. Shouldn't be too tricky,
> provided you're not using equations and such (MathML to Latex would be a
> useful tool, but probably quite time consuming to write).*

I have studied XML for a while both in Java and Tcl. Of course, documentation of XML in Java is better than in Tcl and it is even hard to find out which of TclXML and tDOM I should concentrate on. I have to come up with a set of tags for the XML file so I can try out in Tcl.

comp.lang.tcl: Re: Advice needed: Text editor with tags.

I experimented with XSL a couple of years ago, and I remember that it was a pain to get right. As the tool is supposed to replace work with microsoft word I have to keep it very simple and dead on target of writing modular technical documentation.

>
> *As for Tcl/Tk in this context. I don't believe anyone has written such*
> *an editor. I am working on such a beast, currently entitled ArTcl. At*

I was really hoping that it would not be a beast, otherwise I would already give up...
My thought was to start very simple and then maybe extend features when my users complain.

One font, say courier 10.
All editing functions in the text widget available.
Using selections to mark text and add tags
Have a window listing all tags active at the cursor.
When a part of a text is selected a checkbox list of tags available to turn off a tag.

Most text is plain
There will be predefined sections (modular specification for electronic circuits)
There will be tables with parameters typically
<item>name of parameter</item>
<item>symbol of parameter</item>
<item>condition under which the parameter spec is valid</item>
<item>minimum value</item>
<item>typical value</item>
<item>maximum value</item>
<item>comment to the parameter</item>

As a chip is designed from many modules, I want to be able to mark some of the parameters so that they are propagated one level up to the toplevel specification. I would also want to mark some of the parameters as variables that can be changed on the toplevel and propagated and recalculated in the module specifications.

This sound very complex, and it is also if I want to do it with a word processor like Word. Don't forget that I have to keep a revision track of all changes because I have to document what is in the chip and not what is in my head.

The simplest solution that I could find was to use Tcl and tags in a text widget. The ability to assign as many tags as I need is something I like with the text widget.

> *present, though, it's not much more than a snappy title, and minimal*
> *code. I add bits here and there, but with the current lack of free time*
> *I have to devote to it, it's not going to be ready any time soon. I am*

Re: Advice needed: Text editor with tags.

comp.lang.tcl: Re: Advice needed: Text editor with tags.

- > *developing a CSS parser in pure Tcl, with a view to creating a*
- > *text–widget drop–in replacement which could be used for editing DOM*
- > *trees (e.g. XML documents) with CSS formatting. At present, this has got*
- > *as far as a scanner, which is mostly complete, but has a few bugs left*
- > *to be worked out, and a few bits could do with being refactored for*
- > *neatness and speed. I can email you that code, if you want something to*
- > *work on, and would like to go in this direction.*

I have nothing more than the text widget demo from the standard tcl/tk distribution and a lot of notes and wishes. I have decided to use XML and need to understand how I can utilize features in Tcl for XML

- >
- > *Currently, though, I'd say that LaTeX is perhaps your best bet, if you*
- > *are looking for quality printed output (I'm no expert though). You can*
- > *write LaTeX separating content from formatting, but it's a bit more of a*
- > *pain, and so most LaTeX files contain tons of formatting commands. If*
- > *you go this route, you might want to google for LyX, and maybe also*
- > *TkDVI.*
- >

I really want to use latex. Problem is that latex is not very much liked by people not into programming. Engineers can be very conservative and when they want to use word, there are millions of good reasons against latex. I have made one module in lyx and I had to abandon it simply because even I had to spend too much time reading a manual in order to place a simple table with caption and reference. Typing by hand is too tedious.

- > *Another possibility would be to use some simple markup language, and*
- > *hand–roll a script which renders it into the Tk canvas widget. From*
- > *there, you can go to PostScript via Tk's built in capabilities. This*
- > *could actually be a feasible way to go. Check out*
- > *<http://wiki.tcl.tk/2741> for some experiments I did in this direction a*
- > *while ago.*

Using a canvas was my first idea as I have come to like this widget, but I will then have to take care of a lot of things that exceed my programming skills. One of the text widget examples in the demo triggered my interest and I played around with it for a while with different colors and formats and came to the conclusion that I would probably come to a result quicker with text widget than with the canvas.

I use the tkoutliner for my daily work and found its modified wiki could be something, but again, when I write, I want to concentrate on writing and not on formatting. I have been trying a lot of outliners and lesson learned was that too much formatting distracts from the focus on writing and I waste a lot of time on formatting.

I have lately got some experience with using word and I think with horror how I should educate the users of my templates how to use the

Re: Advice needed: Text editor with tags.

comp.lang.tcl: Re: Advice needed: Text editor with tags.

templates right.

I thought about using Java because I have a nice powerbook to program on, but I also have tcl/tk. Deployment is a problem. I can use Java WebStart, but I would have to install a lot of java virtual machines. As far as I have seen the Starkit can solve my distribution problem. There are many reasons for using tcl.

--

Svenn