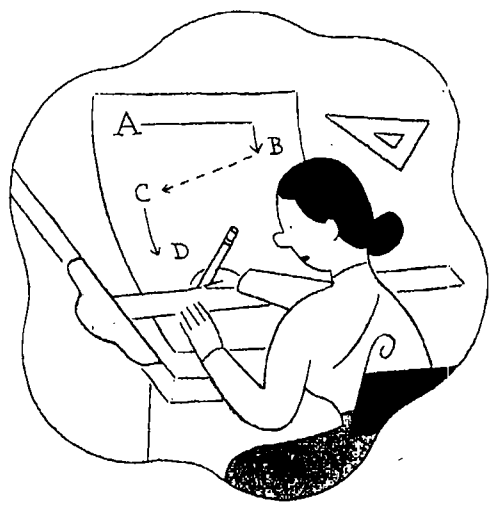
Formatting a Web Page

by Jeffreys Copeland and Haemer

If you work for a living–and that's what this column is about, right?–status reports are a fact of life. But in what format do you submit them? Paper? PostScript? Troff source? A WordPerfect document? The answer, of course, depends on the company. If you've worked for a while, then you've probably run across a handful of different answers.

A few weeks ago, beginning new jobs, we found ourselves faced, once again, with this question. Ever eager to please, we asked our new boss, Steve, which format he wanted us to use to submit our status reports. His discouraged reply was, "You know, that's a good question. We had some meetings to try to resolve this, but we never could agree on a format. Some folks wanted Frame, some wanted TEX, some wanted flat ASCII..."

After a little more probing, Steve said something about how PostScript was a must, but that it might also be useful if whatever we submitted could also be imported into Microsoft Word because our secretary would appreciate that.



Following assurances from a colleague that the easiest way to do this on a UNIX system was to use LATEX and then run a LATEX-to-RTF converter for the secretary, we dutifully set out to learn enough LATEX to produce a decent-looking status report.

After a couple of hours of work, we emailed the LATEX off to the secretary in question, who quickly replied, "I can't use this. Could you please resend it in text mode." A phone call revealed that she meant flat ASCII.

"No problem," we assured her, quickly reaching for our manuals to look up TEX's equivalent of `nroff`.

Jeffrey Copeland (`copeland@alumni.caltech.edu`) *is a member of the technical staff at QMS's languages group, in Boulder, CO. His recent adventures include internationalizing a large sales and manufacturing system and providing software services to the administrators of the 1993 and 1994 Hugo awards. His research interests include internationalization, typesetting, cats and children.*

Jeffrey S. Haemer (`jsh@canary.com`) *is an independent consultant based in Boulder, CO. He works, writes and speaks on the interrelated topics of open systems, standards, software portability and porting and internationalization. Dr. Haemer has been a featured speaker at Usenix, UniForum and Expo Kuwait.*

After another half an hour of frustrated page-flipping, we cornered our LaTeX-literate colleague who explained, "Oh, you can't do that. TeX is just for typesetting."

At this point, you're probably saying to yourself, anyone with half a brain would throw up his hands, fall back on flat ASCII and go back to work. Unfortunately, that would have left us without material for this column.

When in Netland, Do as the Netlanders Do

Intrigued by the sounds emerging from our office, both Steve and our aforementioned colleague wandered in. After some dispassionate discussion, in which they mostly referred to us in the third person, Steve mentioned that no one had ever suggested using HTML for status reports.

"Aha," we said to ourselves, largely because no one would have listened had we said it out loud, "that's it!" It's a safe bet that everyone in the company, from the janitors to the president, knows how to run Netscape. Either that, or they can ask their kids to show them.

The idea of learning yet another markup language wasn't frightening, but once burned, twice shy. Could we make a nicely formatted HTML document that we could both print as PostScript and email to a secretary as flat ASCII?

Well, two out of three isn't bad–it just wasn't the two that we had expected. If you press the right mouse buttons and run the right filters, you can print documents in PostScript and convert them to flat ASCII. Unfortunately, there's no easy way to turn out a nicely formatted HTML document.

It is received dogma that browsers, not users, should have control over the presentation of a document. We are allowed to plead for a different font: "Please emphasize *this* word." But we are told that it is gauche to specify that we want a word in italics or *boldface*.

We are put in mind of our wedding. (Editor's Note: This is the editorial "we." Copeland and Haemer are *not*, in fact, married.) Our sister, a commercial jeweler, volunteered to make the ring. "Go to a local jewelry shop," she advised us, "and get a good idea of what you're looking for–style, color, metal, stone and so on. When you know, call me. Then, I'll just ignore you and make whatever I feel like making." Our mother has a cocktail ring fashioned by our father, the silversmith, which was executed on similar terms, but rejected by the original customer.

There's no easy way to translate your work into a crude ASCII facsimile that you can email to your boss or the department secretary.

We are, of course, control freaks (programmers) and find that sort of attitude in a piece of software entirely unacceptable. Who, after all, is the boss around here?

Luckily, we're not alone. Next to Mirsky's Worst of the Web (`http://mirsky.turnpike.net/wow/Worst.html`), our favorite Web site is David Siegel's (`http://www.dsiegel.com/home.html`). Siegel is a first-class designer, and his site offers lots of advice on how to get your Net pages looking the way you want them to.

Unfortunately, like everything else worth doing, it's a lot of hard work. Siegel gives tips on how to handcraft your Web pages, and one particularly valuable tool, the single-pixel gif, but getting your pages to look good requires real craftsmanship. Let's face it, if we were capable of real craftsmanship, we'd have made our own wedding ring.

The Single-Pixel-Gif Trick

Here's an example of what we mean. HTML lets you mark the beginning of a paragraph with the tag <p>. In the Netscape browser, this produces a blank line. But what if you want the paragraph indented? Put in blanks or tabs?

Nope, Netscape ignores them and left-justifies the paragraph. What if you want two blank lines between paragraphs? Put in extra blank lines? Nope, Netscape ignores blank lines and puts in exactly one blank line for the paragraph. And what if you want something more subtle, like one-and-a-half blank lines?

Siegel's solution is elegant. First, he explains that if you have an array of single-color images, you can use the `<IMG>` tag, with its `height=` and `width=` modifiers, to create solid blocks of color of any size. That done, he puts this technique to work by using solid blocks of *clear* pixels to force indentation and inter-line spacing. He calls this "the single-pixel-gif trick."

Well and good, but you still have to experiment to discover exactly what size blocks you want to carve out on the page. Moreover, all that craftsmanship is directed toward making a good Web page. There's no easy way to translate your work into a crude ASCII facsimile that you can email to your boss or the department secretary.

It's like having a LaTeX document. There's no equivalent of `nroff` that will lay out an ASCII version. (Siegel is, we note both in passing and in jealousy, a student of Don Knuth, the author of TeX.)

This observation, however, sug-

gests a solution. All we need to do is enhance nroff/troff so that in addition to the sorts of output that it already produces–ASCII, PostScript, DVI and so on–it can optionally turn out HTML. Of course, there is the problem of not having the source. But with a little skill and determination, perhaps we could write a backend for groff, called grohtml, analogous to grops and grodvi, that would do the job.

The Plot Twists

Still, perhaps there's something in the idea worth pursuing. What if we wrote an nroff macro package that converts embedded roff commands into HTML to look just the way we want it to? Taking Siegel's recommendation that we supplant the default, user-hostile gray background with a very light, mint green, we can make our macro package automatically begin each document with <body bgcolor='#EEFFFA'> (Warning: You may not have enough color resolution to use this shade. Those of us with color-challenged vision may not notice in any event.) Using Siegel's suggestion that we indent our paragraphs by eight pixels, we can define our paragraph-begin macro like this:

```
.de P
<b>
<img hspace=8 vspace=0
src="resources/dot_clear.gif">
..
```

Siegel tells us that good typography dictates that our lines have 10 to 12 words and that we need bigger margins than Netscape usually gives by default. He then shows a scheme using tables that more or less does the trick but has more details than we want to keep track of. Here again, we can fold a mountain of detail into one or two macros.

Once this is done, we can create HTML documents that follow any guidelines we choose. We've chosen Siegel's because he knows a lot more about this than we do, but you can make your macros do whatever you like. All we do is sprinkle a few simple macros into our text, and let nroff create an elaborately marked-up HTML Web page for us.

To get the ASCII version, we use the same trick that we used to create envelopes for letters in an earlier column. We make a second package using the same macro names that turns out ASCII instead of HTML.

Those of you paying close attention will notice that we've turned the classic UNIX-formatting approach on its head. Traditionally, we would treat nroff/troff as the basic formatter and use specialized front-end preprocessors, like pic, tbl and eqn, to enhance its capabilities for specialized tasks. Here, we are using nroff/troff itself as the front end. When it is generating HTML, we're only using it as a macro processor. We aren't taking advantage of any of its power as a formatter, though we are when we use it to produce ASCII or PostScript from the same text.

Some of you may even see this as deeply warped. Let's face it, we may not have craftsmanship, skill or determination, but...

Elements of Design

Our column's too short to go through all the details now, but it's useful to raise a couple of other design questions before we quit.

• **What should we call our macros?** One of the advantages of using nroff/troff macros is that we're comfortable with the syntax. If you like this approach but tend to use TEX, for example, you'll probably want to write a TEX package that does the same thing but uses TEX syntax.

We've been using troff to format documents for so long that we invoke its formatting commands as casually as we edit files with our favorite text editors. To get the full value out of this familiarity, we should design our macros to mirror the macros we are most familiar with. For us, that's the nroff -mm macro package, so our paragraph macro is .P, our section-header macro is .H, our macro for inserting blank lines is .SP and so on.

This also means we can use the existing mm macros to do most of our work when we need to produce either ASCII or PostScript. At this point, our macro package for producing HTML is under 200 lines long, but our macro package for producing ASCII or PostScript is one, one-line definition–unless you count the thousands of lines of mm macros in /usr/lib/tmac.

• **Paper (and email) documents are linear. HTML documents aren't.** Admittedly, status reports usually don't need anchors or links, but we should permit them where they're useful. There isn't any overlap between the syntax of HTML and nroff, so we can always put in links–or any other HTML tags–by hand. However, we should design a few macros that will make the common cases easier and decrease the likelihood of syntax errors.

For example, after a few reports, we decided we wanted to give each list a table of contents and each item a link to "next," "last" and "top." To put this kind of power into our list, the macros themselves require that we first collect the entire list, extract the item titles and locations within that list and output the annotated result. That's probably more work than it's worth, so we fall back on the UNIX trick of using a preprocessor, which looks for lists and inserts links before handing the resulting source to nroff.

Next time, we'll discuss the preprocessor plus a few other tricks that we have up our sleeves. Meanwhile, if you want a copy of the current version of our package, you can either email us or pick it up from. http://www.qms.com/www/tmac. ▲