HTML Email:
The Perils of Copy 'n Paste (Part II)

March 29, 2011   By Miller Ramos

In my last blog post, I played the part of the copy and paste doomsayer, warning all of the dangers inherent in the act of, well, copying and pasting. In particular, I discussed how, with that simple editing function, invisible metadata that has a rightful place in the source document can find its way into the HTML describing your email communications, (where it has no right to be in whatsoever) producing junk code.

Not to be a hater but here's another reason to be mindful when indiscriminately copying and pasting into your code: junk characters.

Ever read an email or a webpage and then suddenly come across this little guy â€™ mid-sentence? What is that? What in the world is â€™? For that matter, what is â€œ? Or â€¦? Who is doing this?

Sadly, the culprit here is you. Or the person who copied and pasted the content into your HTML, anyway. Those are junk characters, ladies and gentlemen. And I'm sure you've made each other's acquaintance before this little introduction. They are also known to show up as inline empty boxes, solid diamonds with question marks in them, boxes with small letters and numbers in them as well as a variety of other nonsense characters strung together to form equally nonsensical words. That gibberish is your browser trying to make sense of characters that aren't in its ability to display as intended. And it kind of looks like classic comic book swearing, doesn't it? Specifically, what you're seeing is what happens when the browser tries to interpret invalid ASCII characters.

A lot of this happens when trying to read something written in the alphabet (or character set) of another language that the browser or your system doesn't know. Rather than show you nothing, it tries to show you something (something that doesn't make sense and winds up looking terrible, sure, but at least it's trying). Most of the time, though, this happens when

your browser is trying to interpret smart punctuation that has made its way into the text. The most notable offenders are curly double quotes (also known as printer's quotes, smart quotes or typographer's quotes), apostrophes, ellipses and double dashes. These non-standard characters make their way into your text thanks to editors like Microsoft Word that automatically "smarten" your punctuation for you as you type.

Now, here's the thing: this one's pretty sneaky. Why? Gibberish text like this (and there are many different varieties) tends to only appear in the browser and not in the editor. So to find out if this is happening, you should definitely test your mailing before sending it out. (Best Practices says you should, anyway.)

Fixing this is tedious, as you have to locate each occurrence of these rogue characters and retype them as, for example, straight quotes (your keyboard's default) instead of curly quotes. If you're savvy enough, use a global search and replace tool to accomplish the same end. If you're savvier than that, run a script to do it for you. Ideally, once identified, you should go into the code and represent those rogue characters using HTML character entities instead.

To prevent this from happening altogether, make sure that autocorrect/autoformat in your text editor does not automatically replace your use of punctuation with smart punctuation. And make sure that whomever's handing off the content to to be dropped into the HTML does the same.

"Smart" punctuation indeed.