

# Typography

## Typographers' Inn

Peter Flynn  
University College Cork

### Reversed quotes

The pox of reversed quotes (‘) continues to spread. If it isn't checked, we'll have an entire new generation of typographers who believe that 'this' is the right way to implement 'quotes'.

I don't know if it's attributable to ignorance, or if there is a version of some lesser breed of software out there which has implemented it as the default. Perhaps someone who is a more frequent user than I of  $Q-k$   $X-s$  or  $F-r$  and other such tools could check this and let me know.

The glyph is in itself harmless, and as I said when I started this campaign, I first saw it in a book published in the 1970s, so it has a long and dishonorable history. It would be nice to think that its use just displays ignorance or carelessness, but typographers are usually neither ignorant nor careless: there is some deliberate behavior at work, and it goes unnoticed by most readers. I suspect it comes from a mistaken desire for a spurious symmetry rather than anything else, but what mystifies me is how it came to exist in the first place, and where people are finding it in fonts.

The usual pint in Vancouver or other suitable venue for the person who mails me the grossest example of this abuse.

### Tag abuse

Talking of abuse, I shall shortly be relaunching the Society for the Definitive Abolition of Tag Abuse (SDATA). This worthy organisation was set up to campaign for better markup languages in order to relieve authors and editors of the need to abuse existing markup systems, and thus to prevent the more obvious typographic mistakes which arise from ambiguous or meaningless markup.

Tag abuse takes several forms, depending on the language, but the most obvious example perpetrated in  $\LaTeX$  is the use of `\emph` to achieve italics even when emphasis is not the objective. To some extent this is a problem of our own making: for so long we thundered at the poor users 'THOU SHALT NOT USE `{\it }` FOR EMPHASIS, ONLY `{\em }`' that many of them now believe they will be shot at dawn for using the undistinguished `\textit` for italics of

any sort instead of `\emph`. The labs are full of them, and they propagate the myth to every new intake of users.

There is admittedly the advantage to `\emph` that it handles its own context-sensitive font control, appearing in italics within a roman body and in roman within italics, but as this is merely a macro in `latex.ltx`,

```
\DeclareRobustCommand\em
{\@nomath\em
 \ifdim \fontdimen\@ne\font >\z@
 \upshape \else \itshape \fi}
```

I see no reason why it shouldn't be called something like `\romital` and made available for anyone to implement in any circumstance where a context requires a distinguishing font shape.

Despite this kind of misunderstanding, we have been shielded to a large extent from some of the horrors of undistinguished markup: there is far worse outside the  $\TeX$  world. One system I have seen provided perfectly sensibly for `\emph` but covered itself by also providing for `\emph1`, `\emph2`, `\emph3`, and so on, with notes in the specification saying which one was to be used for italics, bold, bold italics, small capitals, etc. This allowed the markup to reflect how the editors wanted the text to appear, but didn't let them specify it meaningfully. They still had to make the decision on which font to use but could not name that reason in the markup.

The whole area of generic markup and meaningful names for things is a two-edged sword for designers. If an author or editor marks some words in italics in a document, does she mean italics *ruat caelum* (come what may), or does she mean italics *mutatis mutandis* (*according to sense*)? The point about markup abuse is that she shouldn't be marking italics in the document at all in this case, but something like `\foreign` instead, and leaving the font decision to the designer in the stylesheet.

We have become so used to commutative font specification, where the surrounding font parameters are inherited, that users now *expect* to be able to get bold italic small cap sans-serif outline swash characters when requested, and it's no use telling them that the font designer only drew swash characters for a few decorative italic capitals. Worse, many DTP systems actually make a feature of providing any permutation of anything vaguely font-like on command.

On the other hand, there's nothing wrong at all with marking decorative italics or bold for what they are, despite the screams of protest from the purists. What's wrong is calling them something that they

are not, like ‘emphasis’. I’ve had users ask me how they can make italics ‘more italic’ because they want increasing levels of emphasis.

As the world is poised to start the slow move away from hard-coded appearance to a more extensible markup system (well, that’s the theory, anyway) it is going to become more important that typographers and compositors are able to untangle the mess left by well-meaning authors or editors unknowingly abusing what they believe to be usable markup. Join now and maybe we can educate them: <http://www.ucc.ie/sdata>.

### Word-swallowing

The other day I was explaining to someone who wanted a mathematics textbook typeset that there were only a handful of fonts which included the mathematical symbols and had math-spaced italic characters (Times, Lucida, Computer Modern, Concrete, and another whose name escapes me at the moment). I must have been less than lucid, because he went away with the idea that L<sup>A</sup>T<sub>E</sub>X could only set in these five fonts!

No damage done, as I was able to explain that L<sup>A</sup>T<sub>E</sub>X could typeset in pretty much anything that was available in PostScript, Metafont, or TrueType: it was only math typesetting that was restricted to the brave few.

The question then arose, could a different body font be used with one of them? Many of you will have seen the effect of setting text in Times and math in CM, and it’s not very pretty, but in this case the math turned out to be less complex than usual, as the book is a remedial work for those who succeeded in skipping math earlier in life. With the concentration on arithmetic and simple fractions, and relatively few symbols beyond +, −, ×, and ÷, it’s perfectly possible to get away with pretty much any suitable book font, such as Palatino.

However, the pretty little PostScript font installer I mentioned last time has taken a major nose-dive. An MS-Windows crash led me to give up my last Microsoft machine and return to Unix, so while I still have the source code for the font installer, I don’t willingly have the platform, and a lot of people have mailed me to ask when it will be available.

The program was written in response to the large number of complaints and requests I get about difficulties in installing Type 1 fonts for L<sup>A</sup>T<sub>E</sub>X systems. Like many long-term users, I spend a small but significant amount of time explaining to others that L<sup>A</sup>T<sub>E</sub>X is not restricted to CM — math mode or not — and Type 1 is still the easiest of the other for-

mats to handle<sup>1</sup> and the typographic facilities provided by pstricks are too useful to pass up. However, too many install-time options, especially for font encodings, made me realise that what most users want is a simple, prescriptive installer which you point at a directory or CD-ROM of fonts and tell it to install selected fonts come hell or high water and not to go asking questions.<sup>2</sup> In its last incarnation it not only did the .afm to .tfm conversion and file-copying, but added the relevant line to psfonts.map, and created the .fd and .sty files in `..\latex\local`, and ran texhash (or equivalent) to update things (it did assume that the user’s installation was TDS-compliant, however).

It was written as a pilot in *Visual DisplayScript* for Windows, which was the only tool I could find at the time with anything like the functionality needed for writing simple windowing utilities, and I mentioned that I was seeking a similar environment for Unix. Several readers pointed me at *Tcl/Tk*, which I was vaguely aware of from earlier attempts but had never managed to get working. The recent versions are hugely improved, and as it is multi-platform, the rewrite of the font installer will be available for Macs, MS-Windows, and X. The bad news is that because of this shift, it won’t be in a usable form for Vancouver, so a large helping of humble pie is my dessert. Sorry.

◇ Peter Flynn  
University College Cork  
Computer Centre, University  
College, Cork, Ireland  
[pflynn@imbolc.ucc.ie](mailto:pflynn@imbolc.ucc.ie)  
<http://imbolc.ucc.ie/~pflynn>

<sup>1</sup> I have had no success in getting TrueType fonts to work in L<sup>A</sup>T<sub>E</sub>X: if someone can point me at a reliable, authoritative, prescriptive, and bug-free document describing the procedure, I’d be very grateful.

<sup>2</sup> The default encoding I use is Y&Y’s LY1, for the simple reason that it’s the only one I’ve found which puts all the characters I want in places where L<sup>A</sup>T<sub>E</sub>X and dvips can find them: if someone can point me at a reliable, authoritative, prescriptive, and bug-free document describing why another encoding is superior, I’d be very grateful.