

One Last Compile...

The RAD Road to Hell

I don't want you to think I'm not grateful or anything, but don't you think sometimes that Delphi is a bit *too* easy? Not all of it, obviously: there are some things which will always baffle me, such as what the icon for Database Desktop is really meant to represent and whether anybody's ever found a genuinely useful role for the Tag property. But I mean the day to day stuff. It's ridiculously simple to throw some components together and have an application half way done before you've even done so much as an Entity-Relationship diagram. Doesn't it make you feel kind of, you know, *guilty*?

I can't help remembering my Computer Studies 'O' Level teacher leaning over my shoulder and telling me sternly, no, it's all very well diving in and patching some code together as quickly as possible, but I really had to learn to do two things. One, I had to learn to structure my code. Two, and even more important, I had to learn the value of *documentation*. (Well, actually there were three things, but we'll pass quietly over what he had to say over my choice of suggestive variable names.) Proper programmers, he told me, *documented* their work. He used to speak like that: you could hear the italics in his sentences.

From that day on, programming was yet another of my teenage activities which came with a bucketload of associated guilt. No longer could I cheerfully spend sixteen hours glued to an my old black and white TV, coding the next bestselling platform game. No, every time I typed a line of code from then on, a voice would whisper in my ear, "You are going to *document* that line later, right?" And I knew I never would, and in the end I would become so demoralised by the thought of my imminent arrival at the portals of programming hell that I would abandon the project half finished.

When I got older, and got a proper job, I resolved to try harder. I became something of a documentation zealot. I would spend weeks alone with my flowcharting template, mapping out the tiniest details of even the most trivial subroutine. Now, I told myself, I was a real coder, just like Mr Hughes said I would be.

But, like with flossing, the glow of self-righteousness slowly faded. My notes became shorter, my variable names once again became rather risqué, and sometimes I even did my flowcharts freehand. The environment I was working in meant that I always had to do some documentation first: it was impossible not to. If you submitted a program for the mainframe to chew its way through, it would sometimes be half an hour before you got your error report back. You simply had to have a clear plan as to how the program was going to work.

Then came Delphi, and even that discipline went out of the window. Now I can write a new application inside sixty seconds. If it doesn't work, who cares? A mouse click, a quick bit of code here, a new component there, hit F9, repeat. A couple of hours later I can end up with a working program totally incomprehensible to anybody except me. Even I'm scratching my head when I come back to it six weeks later: what is the difference, do you think, between `Temp1_Edit` and `EditBox_Temp`? Recently, when I was asked if a client could have a copy of the documentation to look through, all I was able to provide was a couple of scribbled notes on beer mats and a state-transition diagram on a coffee-stained Post-It.

As I said, it's not like I'm not grateful to Borland. I wouldn't want to go back to the mainframe, not for a million pounds. Life with Delphi's great. But, if you're reading this Mr Hughes, I'm sorry, I really am...