

# One Last Compile...

## Getting the right package

It's taken me the best part of the morning, but I can now compile my application again. There were times during that long, long, morning that I began to fear that the magic word 'Linking...' would never again grace my screen.

The villain of the piece is Delphi's package system. Well, we all know the real villain of the piece is me, for being such a clueless waste of space, but I'm sure you'll understand if I blame Borland. They're bigger than me, and probably have broader shoulders.

Let's backtrack a little. Regular readers may remember a few issues back when I reported that I'd had a fit of conscience about all the hints and warnings in my application, and how I'd gone through my code carefully picking them all out. Very satisfying it was too. The trouble is, it sort of went to my head. Once you've done it, you become a bit of a zealot, the sort of tedious person who goes along to Borland User Group meetings and tuts loudly at people who use `with` statements in their code.

Although my application now compiled without a murmur, I noticed gleefully that my various component packages were still a bit of a mess. Whenever I recompiled a package, a huge flurry of hints and warnings would scroll ominously up the screen. You know the sort of thing: *'WARNING: JenniferLopez.pas has been implicitly imported into Package PuffDaddy'* and *'Hint: Your routine at line 68 is total gibberish, and isn't what the customer meant anyway'*.

'Aha', I said to myself (I'm the sort of person who says 'Aha' to himself, which may explain why I'm still single). 'This is clearly an opportunity for a dedicated Delphi purist like myself to further improve the quality of this application, and indeed of all subsequent applications.' For some reason I said this to myself in the voice of JR from 'Dallas', which somehow added enormously to the importance and seriousness of the proposed task.

'Besides,' piped up another voice, Tom Baker from Dr Who, 'it's a far more interesting and satisfying task than the one you're actually supposed to be doing.'

I was just about to start when there was a final interruption. 'Now just hold on a minute', said a third voice. 'Isn't there a danger that you'll mess the whole thing up? Do you really understand enough about packages to do this?'

The third voice sounded like Cameron Diaz and, although it was enough to make me gaze wistfully out of the window for a moment on the off-chance she was passing and fancied sharing my M&S sandwich, she was quickly over-ruled by Tom Baker and JR. Of course I knew what I was doing! Silly old Cameron.

Now, hands up: who really understands the difference between a `.dpk` file and a `.bpl` file? No, on second thoughts, I know you all know. But I didn't.

I think I do now. What I'll say in my defence is that, like most things, you never really know what you need to know until you need to know it, and that was certainly the case about half an hour after I started. Sorting out your packages, no sniggering at the back there, can be an excellent exercise for inducing panic: when you're suddenly told that because of X and Y Delphi is now uninstalling half of your component library, and at that moment your boss phones you up and asks if you can just quickly compile a new version of that mission-critical accounts system, your life really flashes before your eyes.

Eventually, through a lot of trial and error, you figure out what Delphi likes and what it doesn't. Slowly you start to get your components back, usually adding them cautiously one at a time and hoping that Delphi won't object too loudly. (My Delphi talks to me in the voice of John Cleese from Fawlty Towers.) My hints and warning count is now slightly higher than it was when I started, but I think I know enough now to tackle the task again in the not too distant future. If nothing else, I've learned the valuable lesson that I really should pay attention when Cameron Diaz whispers in my ear.