
One Last Compile...

More Joy Of TClientDataSets

Where was I? Oh yes, I was telling you about how great TClientDataSets are. And I have to report that, since last month, I've grown to love them even more.

In particular, I find they are just the ticket for little standalone applications which need some kind of database capability, but where you don't want to be hindered with that old woolly mammoth of the Delphi world, the BDE. Pop one down, add a few fields in the field editor, call the CreateDataSet Method, and it's all done. Saving and loading to a file? No problem, the conveniently titled LoadFromFile and SaveToFile will see you right (all in a nice single file too, none of those horrible .xg0 and .yg0 files that used to accumulate around Paradox). All is sweetness and light so far, then. They have, in all honesty, revolutionised the way I approach database work. But you'll have doubtless spotted in your journey along life's hard road that things are rarely perfect in this world. And, since you ask, here's my list of grumbles.

It lacks a snappy name. (Actually, the name's fine, but I'm bored with typing TClientDataSet, and I suspect you're bored with reading it. Henceforth, it's a CDS.)

My most commonly reported error message has now become 'Cannot

load MIDAS.DLL.' Which actually means 'Can't find MIDAS.DLL'. Which means that somebody's forgotten to copy it along with the .EXE. Now, I admit this is not perhaps the most arduous task in the world. But it's irritating to have to ship an extra file along with your application, not least because it makes me feel like a Visual Basic programmer. And what's the point in making it separate, Borland? (It may be that in Delphi 6 it isn't, but I'd be pleasantly surprised.)

It's only in the Enterprise Edition of Delphi 5, Part 1. Frankly, this is pretty naughty on Borland's part, because they have stated that the CDS is now the approved way of doing cached updates. Which are, let's face it, kind of a fundamental part of any database programming. So all those poor schmucks who can't afford the Enterprise Edition, and surely the majority of users use the Professional version, have been denied the opportunity to future proof their code. Tut, tut.

It's only in the Enterprise Edition of Delphi 5, Part 2. Even worse than the above, the fact that it's only in the Enterprise version means a large number of readers of this fine magazine won't have had the opportunity to find out that I am, for once, talking eminent good sense.

Why the blazes didn't they think of this in Delphi 1? Okay, this may seem a bit picky. But, having seen the light with the CDS, I am now cast into gloom whenever I look at my older database programs. They are knee deep in TTable and TQuery components, and hence all welded at the hip to that blasted BDE. If I ever write a book of programming to-live-by maxims ('Never tell a computer joke in front of clients', 'Girls are rarely impressed by software collections') I shall add this one: 'However many layers of indirection you put in, it's never enough'. There's no easy way of extracting the BDE bits out of my apps, and I suspect the same is true for most people, apart from those smart-alecs who spurned it right from the start. So some lucky soul, me, will spend much of the coming months picking out these components form by form, removing the dreaded DBTables from the uses clause, and using a CDS instead. I think a little bitterness is allowable.

Anyway, enough of my boyish banter. I'm off to DCon now to find out if Borland have put anything worthwhile into Delphi 6. (Completely true story: at last year's DCon I took a seat at the start of a session behind a guy who was clearly determined to stay the distance. He had at least four notepads, six pens, a thermos of coffee and, most frighteningly of all, a roll of toilet paper. I moved to another seat.)