One Last Compile...

Life, the universe and components

If I'm honest (and I like to be honest with you, my dear readers) I have some serious reservations about components.

It seems to be generally acknowledged that components are to Delphi what stock options are to Microsoft employees. You simply couldn't contemplate life without them. And to a point, I agree. I would certainly be lost without the standard VCL components and the various third party goodies you can buy. Where I take issue is with this underlying assumption that everything you do should, in theory, end up as a little component on your palette.

For one thing, it seems unnecessary. The underlying technology in Delphi is so flexible and helpful anyway that facilitating re-use is the least of your worries. If you want to reuse a form, you can simply add it into as many projects as you want. Similarly with units and data modules. Why bother turning it into a little droppable icon when all it's going to do is add to the clutter on your other forms? I mean, where's the consistency in having some forms packaged up as enigmatic blobs in the corner of your screen, while all the others are listed neatly in your Project Manager window? Some of the most infuriating conversations I've ever had with customers have been over this...

Me: OK, here's what your window is going to look like.

Customer: What are those blobby things down there?

Me: Those are what we in the trade call 'non-visual components.'

Customer: So how come I can see them?

Me: But you won't see them when the program's running. They represent other windows which this window will use.

Customer: Ah, so the Invoices window is one of these blobby things.

Me: No, no, that's somewhere else.

Customer: Can we change the colour of the blobby thing?

Which leads me neatly into the crux of the matter: who has the time to design new icons for all these components?

It can take you ten minutes to write a new component, but the best part of a fortnight to come up with an icon design that looks even vaguely relevant and *doesn't* look like it was designed by a colour blind six-year-old. So you end up using the default icon for everything. After all, there is probably no icon on earth that could properly represent `GetOverdueInvoicesForCustomersOutsideLondonForm`. Then, before you know it, half your life is spent hopefully moving your mouse over the palette, reading the hints to try and find what you're looking for. I often buy component libraries on the strength of the icons alone: I may not need the functionality, but I'm strangely moved by the thought of all those lonely hours they spent using the Image Editor.

Remember how one of the things that was always cited as an example of how handy components could be was 'overriding default behaviour'? You would create something with a cutesy name like `TMyFirstEditBox` which would do nothing except set its font to Tahoma Bold Italic. It would do so little, in fact, that you'd never bother backing it up or documenting it. Components lull you into thinking that they somehow exist outside your applications, and aren't in anyway a fundamental part of it. And therein lies the path to what I call 'The Delphi Re-Install Blues', which is brought about by an endless cycle of:

1. Trying to open a project.

2. Being told that a component is missing (with helpful prompts such as '*Would you like to remove the reference? I realise that this would make your entire application non-functional and worthless, but you should keep better track of these things you stupid dork.*').

3. Frantically hunting around your hard disc for likely looking .pas and .dcu files which might enable you to recompile your project and fix the bug that your customer is threatening to sue you over.

4. Repeating 1 to 3 until re-writing the entire project looks viable and attractive.

See what I mean? Components are like booze. You get dependent on them, and then they ruin your life. Just say no (but you should probably say 'yes' to Microsoft stock options).