Richard Kaiser Tuebingen, September 12, 2003
rk@rkaiser.de

To
John Thomas, Borland (jthomas@borland.com)
J.P. LeBlanc, Borland (JPLeBlanc@borland.com)

Dear John, dear J.P.,

As a member of the C++Builder Beta Test program I followed the discussion in the newsgroups and read J.P. LeBlanc's "Open Letter to the Borland® C++ Developer Community" on the community website. I am one of the loyal customers J.P. LeBlanc addresses: I wrote my first book about Turbo Pascal in 1985. I am professor at a German university, freelancing C++ trainer for Borland Germany and for my own company. At the last meeting of the C++ Standards committee in Oxford I was head of the delegation of the German National Standards Institute (DIN). Currently I am working on the English Edition of my book "C++ with Borland C++Builder".

Although many contributions in the newsgroups are unclear and contradictory, these two statements seem to be true:

1. The product line of a VCL based RAD C++ development tool will be cancelled, and there will be no more (or only minor) enhancements, bug removals and upgrades to C++Builder 6.
2. The VCL based C++Builder will be replaced by a completely different tool with nearly the same name.

Let me make some remarks to these points:

1. If the product line of a VCL based RAD C++ development tool will be cancelled, what about all your customers that have built applications that they want to upgrade and enhance? There are lots of companies (I know some of them) that developed big VCL based applications. Do you really want them to learn a new class library, change their sources and test the whole application again? Do you really believe they have time for that? I don't: If they have to learn a new development tool, they will go to learn the Microsoft Visual Studio. With this, they can be sure that they can still use it in 10 years from now.

   If there will be no more enhancements, bug removals and upgrades to C++Builder 6 this is just to say in other words to your customers "It's your fault that you relied on our products". If you want to drive your customers into the arms of your competitors, there is no better way to do this.

   Canceling the VCL based C++Builder reminds me to changing the company's name from Borland to Inprise. This was a great mistake (that I never understood), and you had to undo it (as I always expected). I you cancel the VCL, you will have to undo this too. But at the time you do this, your customer base will be smaller than now, and you will have lost a lot of confidence.

2. Multiplatform is nice, but with a market share of about 90% for Windows, you can't get a big additional market compared to a one-platform tool for Windows. This additional audience seems to be a similar audience (the "X" audience) as that for Kylix/C++Builder, which didn't sell very good. If you provide a multiplatform tool, many customers will assume that it serves the smallest common denominator of all platforms, and it is not optimal for Windows.

   With the VCL based C++Builder you have a customer base. With BCB-MP you can only hope to get one. If that doesn't occur, you shouldn't have burned all bridges behind you.

3. If the VCL based C++Builder product line will be replaced by a completely different tool with nearly the same name this will be confusing to your customers. They will be unpleasantly surprised if they get another tool as the one they expected.

Conclusion:

I will greatly appreciate an additional C++ RAD tool with a new, completely C++ based class library. But

1. I think it will be a very great mistake to cancel the VCL based C++Builder product line. There are too many customers that relied on that. If you take them their tools away they think you will fool them.
2. don't sacrifice the big Windows market for the small multiplatform market. Keep one C++ tool in your product line that is specifically for Windows.
3. this tool should have a different name so that everybody sees immediately that it is something different.

Both products can coexist. So please, please, please: Keep the VCL-based C++Builder alive and kicking for a long time. Complement it with a standards conforming C++ compiler (e.g. like Comeau's) and class library based tool that can be used as easy as the VCL based C++Builder.

It would also be great to have a migration tool that converts VCL projects to new projects. But do not urge your customers to migrate.

Yours
Richard Kaiser
http://www.rkaiser.de/