

A Second Look at Delphi
By Paul Coleman - Compuserv 72056,1044
For the Los Angeles Visual Basic User's Group

Initially I had a quick look at the visual aspects of Delphi. Delphi is impressive. I have done more design and like it even better, following are my recent impressions.

Visual Design - Delphi contains most of VB's third party tools. Following are standard Delphi items that you must buy third party for VB:

- DataGrid component.

- Tabbed Notebook component.

- DataCombo component.

- Most components are 3D.

- Selecting multiple components and right clicking allows you to align and commonly size the components (ala VBAssist).

- The galleries allow you to save common form and project templates.

- If you don't like a component, you can modify it and save it back to their equivalent of the tool box. I have not done this, it seems to take more understanding of Object Pascal than I have developed.

I rate the visual design tools far superior to VB 3.0. I have heard that VB 4.0 addresses some of these issues but I have not seen it and nobody is talking.

Database Tools - Delphi's database tools are a giant step ahead of VB's. There are many more Data Components than VB has data aware controls, so Delphi wins the component/control race. The Delphi data access components are more like VB's data variables, but they act like a data control in that you can bind them to data components. An example is showing a master/detail relational database.

In VB you would -

- Place two data controls one for the master and one for the detail, and link the desired fields.

- You then place code in the Re-Position event for the master to change the SQL to reflect the new master key for the detail data control and refresh the detail data control.

This is not the only way but one that probably uses the least code. You could specify the detail in key order and find the first record that matches the master key or something similar with data variables. The other options would take more code to make sure to maintain the relationship.

In Delphi you would -

- Place two Tables (or Queries) data access components on the form.

- Specify the master in the detail table component.

- Place a data source component on the form for each of the table components.

- Place the data components on your form and link them to the proper data source.

This sounds like more work but it is just placing a couple more components and set their properties and you are on the road. This should be faster because you are using the indexes of the database and don't need to re-query the tables.

Coding - The coding process is very similar to VB. You are using Pascal syntax but that really is not that big a difference. You get tired of the syntax errors for lacking ";" and using "=" instead of ":", but you get used to the syntax quickly.

It is much more difficult to understand the Object architecture. I was familiar with Pascal enough to get into the Uses section to get my units to talk together. I had the hardest time getting Delphi to let me look at component properties on another form (in fact I was not able to figure it out, if was easier to figure out variable scope and pass the contents in the variable). Using the API's is a little tricky as well. Mostly it is a matter of calling them from Pascal rather than C and the lack of examples in help.

There is a learning curve. The syntax is quite easy but the language as a whole is harder to crack.

Auto Scroll Bars on Forms - One question we had at the last meeting was Delphi's ability to place scroll bars on a form when you re-size it at run time. When you re-size a form at run time, the application can either cut off components that no longer fit, or give you scroll bars so you can get to the controls that are not visible. VB simply leaves every thing off the form and does not give you scroll bars. Delphi gives you the scroll bars so you don't lose components that do not fit on the form. The exception is forms that contain auto sizing components.

VBX Compatibility - Delphi is compatible with version 1 vbx's. These are the ones that are compatible with other compilers like C. To test this I tried using VSVBX, the elastic vbx that is popular. I could use the vbx and place the elastic on my Delphi form, but the elastic did not work. This is probably a minor problem if third party vendors make VCL's or if Delphi supports OCX's in the future.

Sample Project - I set upon a task of making a simple project using only standard components with both VB and Delphi. The project is a VB or Delphi launcher. It stores a list of projects in an ini file and will launch VB or Delphi specifying the project you want. This saves you time loading the application and then loading the project you want. It took me about the same amount of time. This is really amazing considering the language learning curve. VB would have taken much longer than Delphi but I cheated a little. I always start with a template application, this saves coding a default menu and toolbar (Third party tool). I uploaded the startup.exe to the LAVBUG BBS.

Wish List - Like everyone else I will not be satisfied until I find a development tool that writes

the code I want by itself. The following are some things I would like to see in Delphi.

Better help with more examples that can be cut and pasted into your application. I assume that the released version has better help than the Field Test version.

One real gripe with all the help systems is that it is really easy to find help once you know what you want. If someone could figure a way to overload the help system so you could search on the result rather than process. That is to say, it would be nice to search on "Passing Strings to API" rather than looking up the API and knowing that you need null terminated strings and finding the function to convert a Pascal string to a null terminated string.

It would be nice to have a little more help with setting the scope of a variable and components. When you create a new form Delphi should automatically insert the uses clause into the current project or at least prompt you through the process. The same should be true of the scope of variables and components.

The Database Forms Expert is nice but would be nicer if I could add the database components to my form. I wanted to put the Master information on one tab in a notebook and the details on another. I did a lot of cutting and pasting (caused one of my stack overflow errors). If I could place edit boxes for all the fields in a table in the current container, I could save work.

Conclusion - Delphi is a very good tool. You can quickly prototype and have a more professional feel to the tools. I crashed Delphi a time or two. I am using a demo version of the field test version of Delphi. I don't know if I was just trying to do much but I did cause a stack overflow and Delphi abruptly crashed and lost my work. I am satisfied with these results in a new product. I think every one needs to consider Delphi in their tool box.