

From Visual C++ Application to Native Mac in 90 Seconds

By Jason Vokes

Embarcadero Technologies

April 2012

Americas Headquarters

100 California Street, 12th Floor
San Francisco, California 94111

EMEA Headquarters

York House
18 York Road
Maidenhead, Berkshire
SL6 1SF, United Kingdom

Asia-Pacific Headquarters

L7. 313 La Trobe Street
Melbourne VIC 3000
Australia

FROM VISUAL C++ APPLICATION TO NATIVE MAC IN 90 SECONDS

WHY YOU SHOULD BE INTERESTED IN WORKING ON BOTH WINDOWS AND MAC

Have you noticed an increased presence of Mac most everywhere recently? This is not just true for coffee shops and public transport, but ever more in the workplace too. Industry analyst Forrester says that almost half of enterprises (with 1,000 employees or more) are now issuing Macs to employees – and they plan a 52% increase in the number of Macs they issue in 2012. Apple's Mac sales grew 66% in the enterprise in the first quarter of 2011, and later in the same year, Mac sales growth was reported to have outpaced the PC market by 20:1 according to industry analyst Gartner.

PRESSURE ON IT AND DEVELOPER TEAMS

This change makes for even more pressure on the IT department and especially for developers, who now need to provide quality software for an increasing number of professional Mac users. All of this on top of continuing to innovate in the business and support existing Windows users, and that is even before considering tablet and mobile device support.

WHAT ARE THE OPTIONS? IGNORE? WORK TWICE AS HARD? ...OR GET SMART?

Firstly, you could choose to ignore the need to support other platforms and just continue as a shop that only targets the Windows platforms. This means that you will still be supporting the majority of users, you won't have to learn anything new and can continue with business as usual. However, ignore these emerging platforms and devices at your peril. You will be ignoring a significant minority – often comprising of many key workers and decision makers and the overall business will be missing out on a growth opportunity.

Secondly, you could develop specifically for both Windows and Mac. This would mean great native support, and fully embracing the business opportunities that arise from supporting these platforms, but this comes at a cost. It is going to be costly to set up a parallel development team alongside your Windows team involving new spend on hardware, software and skilled developers. Also, once you have everything setup and ready, running and managing both initiatives will cost more than twice that of a single Windows team. You may even see a compromise in features or user experience in order to have a comparative

product on each platform. So, for most organizations, even with highly skilled developers and the best of intentions, who can realistically get the time and resources to be able to completely retool, reskill and then run/write two versions of the same project in parallel?

There is a third way and C++ developers have a particular opportunity to make the most of their existing skills and code to achieve this. Embarcadero, independent developer tool experts, recently introduced FireMonkey, a native application framework included in the RAD Studio family of tools that supports the C++ language. FireMonkey enables Windows-developers to create rich business applications with native performance, not only for Windows, but also for Mac from a single codebase. What is more, if you have iOS requirements, you can also target iPad and iPhone devices too. Even though FireMonkey does not support frameworks like MFC, you can still take significant amounts of existing C++ code and use it in projects and applications that then run natively on both Mac and Windows.

FireMonkey is able to natively target multiple platforms by taking the low level tasks that an application would otherwise rely on the operating system to perform or provide. You can find out a lot more specific technical information and how you can try the power FireMonkey in RAD Studio, from the references at the end of this article.

RUN IT ON WINDOWS, RUN IT ON MAC

So let's see what would be involved in taking a C++ Windows application over to Mac using FireMonkey. If we take an illustrative example of an existing Visual C++ project and consider the common strategy of separating business rules from the user interface, we can very quickly take code over to a FireMonkey project – within the Windows based RAD Studio IDE and provide the same application compiled natively and running on the Mac.

Figure 1 shows a common Customer/Order/Product application that loads data into separate linked-lists and then works on them or displays them from there. We have several classes contained in class declaration and definition files Customer.hpp/.cpp, Order.hpp/.cpp and Product.hpp/.cpp. These are included in the solution and are all called from, but are not part of the user interface code.

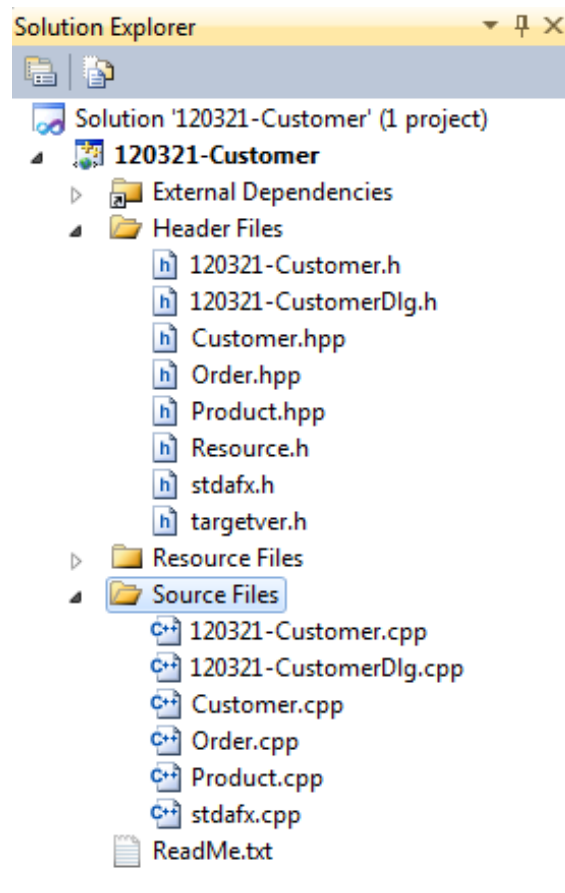


Figure 1: Visual C++ Solution with Customer/Order/Product classes

Figure 2 shows a simple operation on these classes as the application loads and organizes the data. We take each customer record, and show every order for that customer, which in turn, finding each order-item record and loading up corresponding product information, then displays this in an organized – if straightforward – fashion.



Figure 2: Visual C++ Application, Running on Windows Displaying Customer/Order/Product Information

The first stage of getting the application over to Mac is to migrate code over to a C++ HD FireMonkey Project in RAD Studio. Figure 3 shows what the FireMonkey project would look like. Notice the same class files have been included directly.

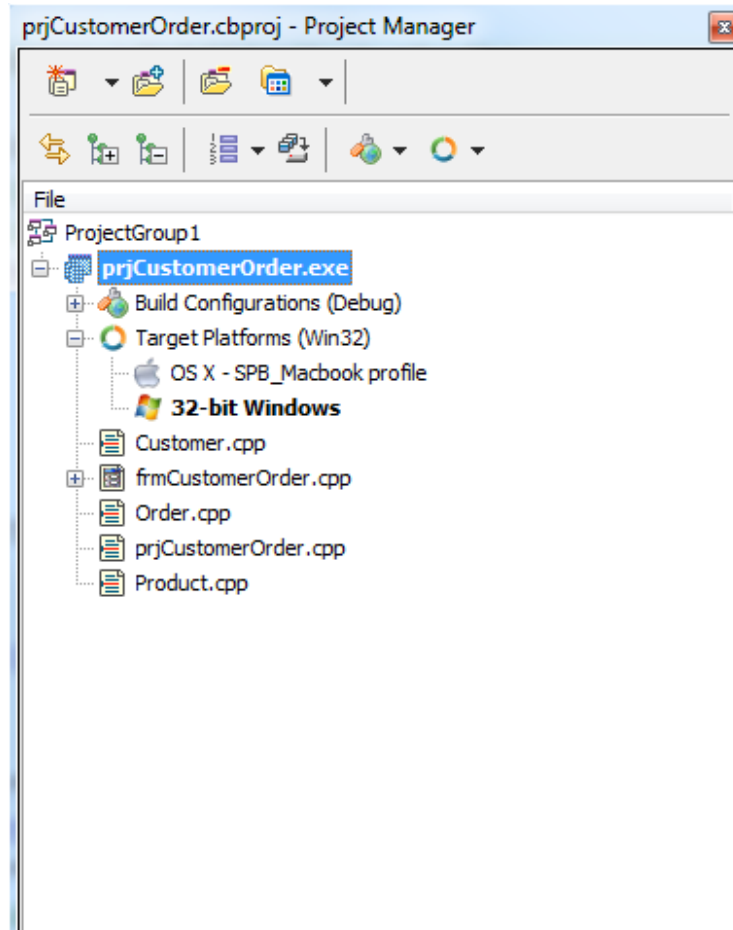


Figure 3: The Customer/Order Application as a FireMonkey Project

Having added the class files into the project, we need to make sure that the event handler code is brought over and ready to be compiled for Mac. In the case of this application, this is the OnClick event for the "Load Customer Orders" button that you can see in Figure 1. You can see this code in the Appendix.

There is a lot of similarity between the UI code for MFC projects and FireMonkey, but you will need to watch out and adjust handling of objects according, such as the MFC equivalent of this line...

```
ListBox1->AddString( (LPCTSTR) strTemp );  
...would be this...  
ListBox1->Items->Add( (const char*) strTemp.c_str() );
```

Once you have your project compiled on Windows as a FireMonkey project you can test by running this on Windows first.

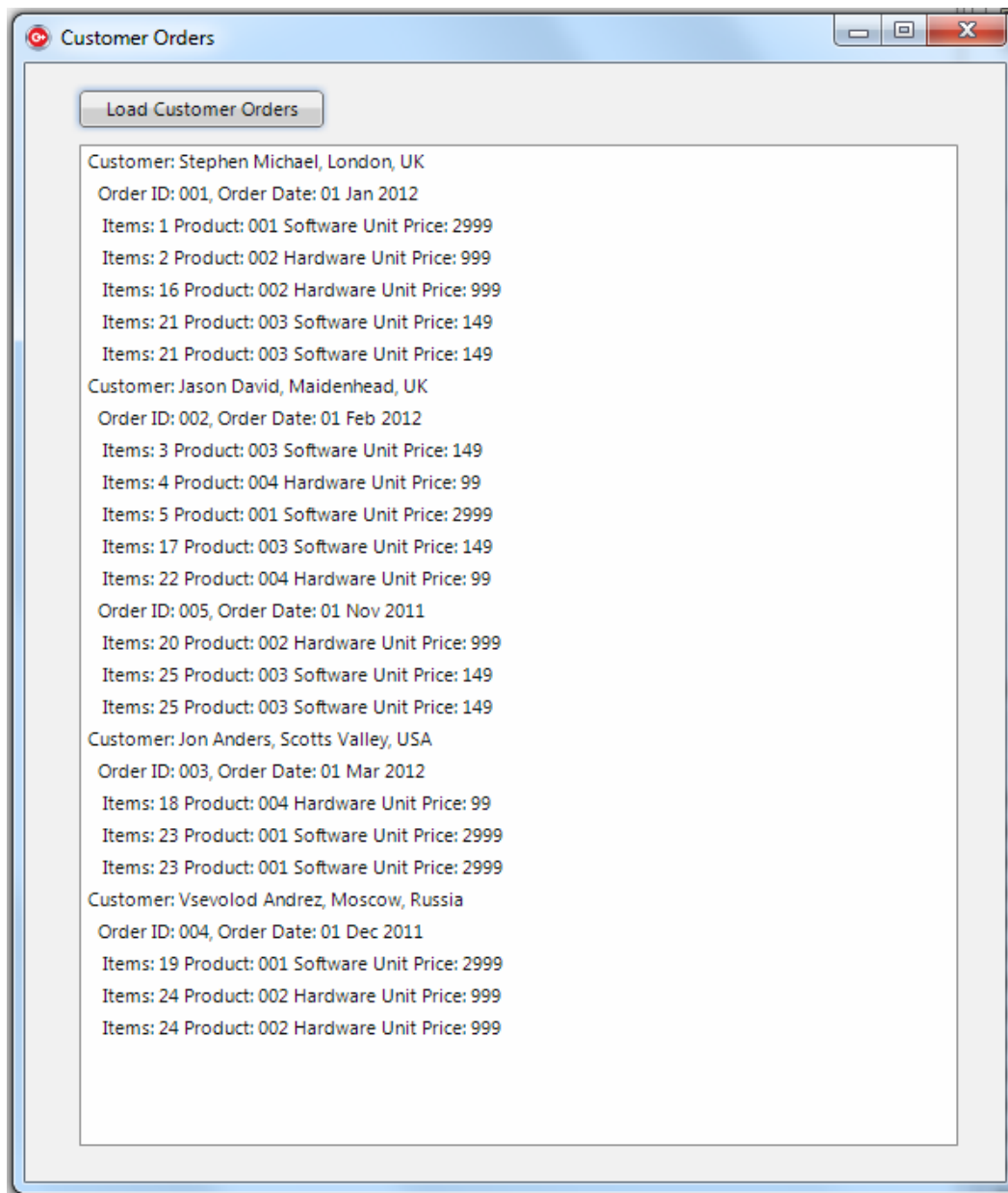


Figure 4: The Customer Order FireMonkey Application, First Running on Windows

With the application compiled and now running as a FireMonkey Windows application, we can develop and test the application as much as we require, before switching the Target Platform (as can be seen in Figure 3) to compile and run the native application on a target Mac machine.



Figure 5: The Customer Order FireMonkey Application Running on Mac

By the same measure, and in a following article, we can show how you can also use code that compiles under Visual C++ and re-purpose this in an iOS application that would run on iPad or iPhone. Embarcadero have also announced that the FireMonkey platform will be supporting Android and will also have the capability to target other platforms as they emerge in the market.

SUMMARY

In this short article you have seen that there is not only a huge opportunity to reuse code on other platforms, and to do this in native applications that make the most of the target platform, through the FireMonkey Application Framework. You can also achieve this in a much more straightforward way than previously possible and even share code, if not the whole code-base, between those platforms.

REFERENCES

Find out more about FireMonkey here...



<http://embt.co/firemonkey>

www.embarcadero.com/products/firemonkey

Download the power of FireMonkey in RAD Studio Today!

<http://embt.co/try-radstudio>



Embarcadero Technologies, Inc. is the leading provider of software tools that empower application developers and data management professionals to design, build, and run applications and databases more efficiently in heterogeneous IT environments. Over 90 of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero's award-winning products to optimize costs, streamline compliance, and accelerate development and innovation. Founded in 1993, Embarcadero is headquartered in San Francisco with offices located around the world. Embarcadero is online at www.embarcadero.com.