REALbasic versus Apple's XCode

By Jerry Lee Ford, Jr.

November 2006

Just as REALbasic competes with Microsoft's Visual Basic as a tool for application development for Windows software, REALbasic also has plenty of competition in the Apple world. One competitor of particular significance is Apple's XCode. Like REALbasic, XCode is a tool suite and integrated development environment for software creation on the Mac OS X operating system. Apple developed XCode to assist programmers in creating executable applications that run natively on PowerPC and Intel based Macintosh systems.

There is no doubt that XCode provides an effective and powerful software development platform. Programmers with a strong Macintosh background will not doubt want to give it serious consideration. However, unlike REALbasic, XCode's capabilities are strictly limited to Mac OS X development. Unlike REALbasic, XCode in not designed to support the development of applications on previous versions of the MAC OS. As of the date that this book was written, the current version of XCode was XCode 2.1.

XCode's focus is on providing a powerful software development environment that facilitates application development using any of an assortment programming languages. XCode is also designed to provide programmers with access to the latest Apple technologies. The programming languages that are supported by XCode include:

- C
- C++
- Objective-C
- Java
- AppleScript

Integrated Development Environments

XCode provides Mac OS X programmers with a sophisticated IDE. From this single IDE, a programmer can develop applications using any supported programming language. This way, once the programmer has become comfortable using the IDE with a given programming language, it is somewhat easier to learn another XCode supported programming language because the IDE stays the same. As you would expect, the IDE provides built-in editing capabilities that assist the programmer in developing applications.

XCode provides everything needed to build a user interface, edit application code and to test and debug program code. XCode provides the ability to graphically represent and view applications, allowing software developers to visually examine the overall structure and design of their applications. XCode also provide a graphical remote debugging feature that allows a programmer to execute an application on one Macintosh computer while debugging the application from another Macintosh computer.
XCode's IDE provides search and navigation capabilities. In addition, XCode provides templates for creating applications, applets, command-line utilities and so on. XCode also provides an integrated Apple Reference Library that includes all of Apple's developer documentation as well as access to Apple's online documentation.

Just like with the REALbasic IDE, XCode's IDE also support sophisticated debugging capabilities, supporting features like breakpoints and flow tracing.

XCode's Integration with Frameworks and Virtual Machines

XCode supports application development for the Mac OS X through integration with different Macintosh specific software development frameworks, including Carbon, Cocoa. XCode also integrates with and supports the Java Virtual Machine.

Carbon is a framework make up of a collection of application program interfaces (APIs) that supports the development of applications for the Mac OS X operating systems using C and C++. Cocoa is a framework designed to support application development on the Mac OS X using the Objective-C and Objective-C++ programming languages. The Java Virtual Machine is a software-based virtual computer that runs on top of a host computer's operating system and acts as if it were its own computer. This allows Java applets to be run independently of the host operating system.

XCode's Key Features

XCode is designed to make application development of Mac OS X applications easier. One more way that it accomplishes this is using predictive compilation, in which XCode takes advantages of lull periods to work on advance compilation tasks. XCode also supports distributed builds, in which other Macintosh computers on the network are leveraged in order to spread and speed up build tasks.

Like REALbasic, XCode also supports remote debugging and other features such as Fix and compile in which the programmer is able to pause the execution of an application while testing it and then make a change to the application in order to fix any problems that may be discovered. XCode also assists programmers in streamlining their applications by helping to produce leaner code. It accomplishes this using a technique known as dead-code stripping, where any unnecessary code is automatically removed from the application when it is compiled. This feature helps reduce application size and can even reduce application memory requirements.

How does XCode Stacks Up Against REALbasic?

As you can see, XCode provides a powerful and robust integrated development environment for the development of software designed specifically for the Max OS X operating. However, it falls short of REALbasic in many ways. For example, while REALbasic is based on a easy to learn yet powerful implementation of BASIC, XCode only support AppleScript, Java and variations of C and C++.
AppleScript is not robust enough to develop interactive graphics based applications and the other XCode supported languages are much more involved and difficult to learn. Most people find Java a challenging programming language to master. In order to learn Java, you must also learn to work with the Java Virtual Machine. Working with Carbon and either C or C++ means that you must know how to work with both the framework and the programming language. The same thing is true for Cocoa and Objective-C and Objective-C++.

The Java Runtime Machine and the Carbon and Cocoa frameworks all introduce extra overhead onto applications. REALbasic, on the other hand, is easier to learn. It doesn't require you to learn how to work with and master a 3rd party virtual machine or a framework. REALbasic creates operating system native applications that can run directly on the operating systems without the resources provided by a virtual machine or framework.

XCode provides Macintosh programmers a solid foundation for applications development provided that you are solely interested in creating applications just for the Mac OS X. XCode falls short is in the area of cross-platform development. Programmers who develop an application using XCode and one of its supported version of C or C++ must be prepared to rewrite their applications in order to port them over to Windows or Linux. While versions of C and C++ exist on both Windows and Linux, the programmer needs to be prepared to deal with differences in the way the languages are implemented on other platforms. In addition, the programmer much also be prepared to learn to work with different IDEs on each platform to which the application will be ported. These factors typically result in a substantial learning curve and significantly elongated deployment period as applications are reworked after being ported over to the new operating environment.