

According to Hoyle...

Cross-Platform Software Development from a Macintosh Perspective: Basic Compilers (Part i)

by Jonathan Hoyle

jhoyle at maccompanion.com *macCompanion* May 2006

We have been covering cross-platform software development options for these past 8 months:

- **Intro:** http://www.maccompanion.com/archives/september2005/Columns/According_to_Hoyle_1.htm
- **Qt:** <http://www.maccompanion.com/archives/october2005/Columns/AccordingtoHoyle.htm>
- **wxWidgets:** <http://www.maccompanion.com/archives/november2005/Columns/AccordingtoHoyle.htm>
- **CPLAT:** <http://www.maccompanion.com/archives/december2005/Columns/AccordingtoHoyle.htm>
- **REALbasic:** <http://www.maccompanion.com/archives/january2006/Columns/AccordingtoHoyle.htm>
- **Runtime Revolution:** <http://www.maccompanion.com/archives/february2006/Columns/AccordingtoHoyle.htm>
- **•AMPC:** <http://www.maccompanion.com/archives/march2006/Columns/AccordingtoHoyle.htm>
- **Java compilers:** <http://www.maccompanion.com/archives/april2006/Columns/AccordingtoHoyle.htm>

Last month, we reviewed the state of Java compilers. In this month's column, we review the state of Basic compilers for the Macintosh.

Disclaimer: "BASIC" vs. "Basic"

Before I receive an inordinate number of emails on this point, let me say that I am intentionally spelling the programming language *Basic* without all caps. According to an older convention, programming languages which are acronyms or abbreviated merges are supposed to be written completely capitalized, whereas names which are not are just initial capped. For this reason, *Pascal* and *Java* are not fully capitalized, whereas *FORTTRAN* (FORmula TRANslation) and *COBOL* (COmmon Business Oriented Language) are supposed to be. However, in this day and age, all capitalized words are eyesores and appear as shouting to those using a computer. Furthermore, it is not a convention universally held; for example, when was the last time you capitalized SCUBA (Self Contained Underwater Breathing Apparatus) or LASER (Light Amplification by the Stimulated Emission of Radiation)? It is also a bit silly to argue that *Basic* is a real acronym: it supposedly stands for *Beginner's All-purpose Symbolic Instruction Code*, yet clearly (as with so many other "acronyms"), the acronym was chosen first, and words are found to retrofit in to make the acronym look good. (After all, shouldn't the language have been called *BAPSIC*?)

I will use the convention of capitalizing only the first letter of a programming language name, which is pronounced as a word, but I will all-cap a language name if it is pronounced by reading its letters. So *APL* (A Programming Language) and *PHP* (PHP Hypertext Protocol) remain in all caps, but *Prolog* (PROgramming in LOGic) and *Lisp* (LISt Processing) will not. I will, however, honor the spellings of actual product names, which have their own rules about spelling, which we see with: *FutureBASIC*, *Visual Basic* and *REALbasic*.

A History of Basic on the Macintosh

Before embarking on a list of Basic development tools and reviewing them, it is particularly valuable to briefly cover the history of Basic implementations on the Macintosh. A historical background will give the reader a bit of insight as to how and why the Basic language tools are in the state they are today. Furthermore, it is a very revealing look at how Mac users of the past were treated and mistreated. The story of *MacBasic* in particular is a powerful one, and it should give you a better appreciation of how these tools came to be (or in this case, didn't).

The Fascinating Story of MacBasic

When Apple engineers were designing the Macintosh, they wanted an implementation of the Basic programming language which was powerful yet easy to use. They wanted to truly empower the Mac user, not just give him the same old console-like Basic, such as Microsoft's implementation on the Apple II. The company's guru of Basic programming in 1984, Donn Denman, poured his heart and soul into the project, creating *MacBasic*, an advanced, object oriented version of Basic, one which could even control many aspects of the Macintosh GUI. It was a revolutionary feat of engineering and received rave reviews from all who saw the demos. By the spring of 1985, beta testers were using it heavily, and it was highly anticipated by all for a summer release. Unfortunately for Apple, Microsoft CEO Bill Gates happened to witness one of these demos.

By this point in time, Microsoft had already ported their mediocre console-like Basic interpreter to the Macintosh. As a product, *MS BASIC* couldn't hold a candle to *MacBasic*, and Gates knew it. He was determined to stop *MacBasic* from shipping. Even as early as 1985, Bill Gates knew that he could leverage one monopoly to help create another. As it happened, the Apple II's Basic (built into the computer's ROM) was written by Microsoft and its license agreement with Apple was up for renewal that coming September. So Gates blackmailed Apple: If Apple doesn't kill the *MacBasic* project, Microsoft would refuse to renew its Basic license for Apple II.

Now at the time, the Macintosh was still a very young product, and the vast majority of Apple's revenue was still coming from Apple II sales. Although everyone knew that the Macintosh would eventually overtake the Apple II in sales, the company in 1985 still relied very heavily on the health of its older Apple II line. Apple President John Sculley, knowing that he could ill afford to jeopardize the Apple II community, capitulated to the demands. In exchange for Microsoft renewing the Apple II Basic lease, Sculley was forced to sell *MacBasic* to Microsoft for...(get this)...\$1.

That day, Denman was told by Apple management that the *MacBasic* project was being terminated immediately, and that he had to destroy all existing copies, the source code and any documentation. They refused to give him an explanation, despite his desperate pleas to know why. The stunned Denman watched as several years of his painstaking work got deleted. Heart-broken, he left the building on an emotional and ill-advised ride on his motorcycle. On his way home, he was involved in an accident, which totaled his bike (fortunately, Denman survived with only minor injuries).

Apple attempted to retrieve all the remaining copies of *MacBasic* it had distributed to beta testers, but when word got out what was happening, the beta testers refused to turn in their disks. The *MacBasic* beta became widely pirated and circulated amongst developers, and it remained in use for some time. Surprisingly, the two planned books on *MacBasic* continued into publication, and, despite all else, sold well.

As for Microsoft, the unimpeded *MS BASIC* became the dominant seller of Basic on the Macintosh for the next several years. With *MacBasic* now the legal property of Microsoft, many of its ideas and unique features were cannibalized to be used for the creation of *Visual Basic* years later. But that is a story for another time.

The Basic Wars

In the early years, the Macintosh was the battleground of several Basic development tools, all vying for users on the new computer. In addition to *MS BASIC*, there was *True BASIC* by Kemeny & Kurtz (the inventors of the language), *Z Basic* from Zedcor, *Softworks BASIC*, and *PCMacBASIC* from Pterodactyl. Each of these products were cross-platform right from the start, with support for DOS, Apple II, Atari ST or Amiga by at least one of them. *True BASIC* was the most similar between platforms and was an excellent learning tool, but was interpreted not compiled. Reviewers favored *Z Basic*, with its fast execution speed and compilation time. *MS BASIC* and *Z Basic* were best for application development, as they gave Toolbox access to the developer.

As there were far more Basic products than Mac developers to support them, the Basic Wars were fought passionately by the aforementioned vendors. *Softworks BASIC* and *PCMacBASIC* were the earliest casualties. *MS BASIC* held the marketshare lead. Although not as polished as Microsoft's product, Zedcor's *Z Basic* became the darling amongst serious developers, whilst *True BASIC* held the schools and still eeked out an existence in the user market. This tenuous equilibrium continued through the introduction of the Mac Plus, the Mac II, the extended Mac II line and onward. With Apple's delivery of *HyperCard* however, many potential Basic programmers defected for this simple, easy-to-use utility. A crisis was headed for Basic tool developers, and only one company would come out ahead.

QuickBASIC for the Macintosh

As the 1980's drew to a close, Microsoft was about to take a commanding lead, turning the Basic Wars to a Basic Massacre. Leveraging some of the powerful features of the now defunct *MacBasic*, Microsoft evolved its Basic into *Microsoft QuickBASIC for the Macintosh*. *QuickBASIC* became the premiere platform for Basic developers. It shared little in common with the PC version, so it was only nominally cross-platform. However, building utilities easily and quickly was its greatest strength, as it was a very intuitive environment to work with. Microsoft was able to make inroads across the board, causing *True BASIC* to shrivel down to only the educational market. *Z Basic* managed to survive but with a vastly reduced marketshare. By System 7's introduction, the only Basic anyone was talking about was *MS QuickBASIC*.

However, after having taken such great pains to eliminate the competition on the Mac, Microsoft did not seem interested in supporting the user base that they had fought so hard to win. *Microsoft QuickBASIC* received only the barest incremental upgrades, with no new features or capabilities added. It was almost as if running out the competition was an end in itself, as Microsoft ended all further development on this product. Whereas the PC version of *QuickBASIC* would be evolved into *Visual Basic for Windows*, the Macintosh version lay stagnant. The last update released by Microsoft was in April 1992 with version 1.00e for System 7 compatibility (yet never becoming 32-bit clean).

Back to the FutureBASIC

It is said that nature abhors a vacuum. Apparently, Macintosh Basic marketplace feels the same way. The void left by Microsoft in the early '90s opened the door for one company to step in and give the user base a product that they yearned for. And that company was Zedcor.

Z Basic had always been the most efficient and optimal Basic, but Zedcor took this product to all new levels of capability as *FutureBASIC*. Dropping compatibility with other platforms and concentrating on the Mac only, Zedcor was committed to create a Basic that would be embraced by the entire Mac development community. Where *QuickBASIC* was a great environment for Basic programmers, *FutureBASIC* was designed for general software development as well as handle old style Basic coding. To allow for an easier transition, Zedcor even created a *QuickBASIC* conversion utility. Though not as fully featured as Microsoft's *Visual Basic* on Windows, *FutureBASIC* represented the best Basic development environment available on the Macintosh, up to that time.

The Macintosh was in quite a volatile state in the mid-1990's with the introduction of the PowerPC. Software developers were scrambling to access newer tools which took advantage of this new chip. Zedcor's early enthusiasm began to wane, so Andy Garipey and Chris Stasny (the original implementers of *Z Basic*) left and took *FutureBASIC* with them and produced *FutureBASIC II*. *FB II*, now sold by Staz Software, continued to make improvements. The next version, written as *FutureBASIC^3*, was PowerPC-native.

Although *FutureBASIC* commanded the lead in marketshare through most of the 1990's, it was not the only Basic tool available. Mainstay introduced one called *VIP BASIC*, which became somewhat popular in the mid-90's. Unfortunately, it was not a native code compiler, nor did it support FPU's. It created large executables due to the attached runtime, and never acquired sufficient mindshare. Its last version was 2.5 in 1997. One which did receive some notoriety was *Visual MacStandard Basic* (first by ZCurve, later by Squirrel Software) primarily due to its incredibly low price of \$29. Being introduced just around the time *VIP BASIC* was becoming desupported, *Visual MacStandard Basic* offered full PowerPC and 68K compilation, a GUI builder, and integration with C/C++ code. Unfortunately, it was still quite buggy and wasn't fully robust. And after what came next, *VMSB* would not last beyond 2001.

REALbasic Takes the Stage

Prior to the introduction of Windows 95, the Basic programming language was not taken seriously by many. Some viewed Basic as an introductory language only, designed for hobbyists and students only, not for serious software developers. One of the more mean-spirited quotes came from the Turing Award winning computer scientist Edsger Dijkstra, who made the public statement: "*It is practically impossible to teach good programming to students that have had a prior exposure to Basic: as potential programmers, they are mentally mutilated beyond hope of regeneration.*"

This unfair characterization was largely alleviated when Microsoft introduced *Visual Basic for Windows*. VB is one of the most powerful easy to use integrated development environment for any programming language. With the advent of *Visual Basic*, the Basic programming language finally started receiving the respect it deserved. Unfortunately for Mac users, Microsoft had no intention of porting VB to the Macintosh as it had with QB. Even with Steve Jobs' triumphant return to Apple in 1997, the situation remained unchanged, as Microsoft held all the power in their relationship. For example, one of Jobs' first major decisions after his return was to capitulate to yet another of Bill Gates' monopolistic demands: either make *Internet Explorer* the default browser on the Macintosh, or Microsoft will kill *Office for the Mac*.

What Apple could not do, Andrew Barry could. Andrew created a shareware product in 1997, then called *CrossBasic*, which was a fully object oriented language that would create 68K or PowerPC apps, but also had the ability to compile to Java applets. Furthermore, it had a very RAD-style GUI builder which made it easy to design applications visually. This was the first product on the Macintosh that was functionally comparable to *Microsoft Visual Basic for Windows*. When Geoff Perlman discovered and used *CrossBasic*, he was so impressed that he approached Andrew with the intent to commercialize the product. Renamed *REALbasic*, it has taken on the Mac development community by storm, as no previous Basic implementation has. Although it no longer compiles to Java bytecode, today it is hosted on (and compiles to) Mac OS Classic, Mac OS X, Windows and Linux. It can truly be thought of as a cross-platform version of *Visual Basic*.

Next Month

Having seen how Basic has evolved from the past, we will review the current state of Basic compilers as they exist today on the Macintosh. Since we already reviewed *REALbasic* in a previous article, we will look at the remaining alternatives and look at their relative strengths and weaknesses. These include: *FutureBASIC 4*, *True BASIC*, *Chipmunk Basic* and others. Stay tuned!