

Miscellaneous Ramblings

This is the best part of the manual in some ways. I can take a more personal approach to writing, and not worry about the spelling and grammar or whether I'm explaining things clearly enough. Some sections of this chapter may hold little or no interest to you, but if that's the case, skip them.

Historical Perspective

BBEdit started life as existence proof that a multi-window text editor could be written in one simple, well-organized source file. It was originally going to be a TextEdit-based demo program. It soon became clear that TextEdit wasn't going to work, mainly because of its 32K limitation on the size of text files and because of its abysmal performance when working with files above a certain small size.

By the simple expedient of a global search and replace, BBEEdit was laboriously converted (it took about fifteen minutes) to use the old "CAPPS' Editor Toolkit". Capps had been produced by the former THINK Technologies, which had since "merged with" (been purchased by) Symantec, who had then discontinued the product. At this point, BBEEdit was named, and still fit in one source file. The twin B's at the front came from the fact that this new editor was "bare bones". You could edit and print text files, and there was a simple "Find" command, but that was about it.

Enter the rest of the technical staff at THINK Technologies, uh, Symantec's Language Group. These guys wanted a decent text editor. THINK C had a good text editor, but it required that a project be open before text files could be opened, and so wasn't good for general-purpose editing. The "PEdit" sample program which was part of the Capps package was good, but couldn't open more than four text files at once, and had a clunky searching interface. It was clear that the demand was there.

BBEEdit 1.0 was completed in Fall 1989. It was fairly simple: it opened an unlimited number of files, subject to available memory, and utilized the appropriate portions of the Capps library to provide large-file editing and multi-file search. It was decent.

Still, people needed to be convinced to use it. What I needed to do was to come up with a set of features that no other editor had, so that people would take one look and be converted. World domination was the goal, and BBEEdit was the vehicle. By that time there were other free or shareware text editors out there. By closely examining them, I came up with some good idea of what to not do, and what other people had done right. I also talked to the engineers at THINK, who comprise some of the best talent I've ever been privileged to work with. By the time I was done, I had a program that had something in it for everyone. One guy wants to print in Courier 6 but edit in Monaco 9? It's in there. The optimizer guy wants batch reporting of multi-file search results? Like Prego™, it's in there.

By the time I was ready to show BBEEdit to the world, the term "bare-bones" elicited a certain amount of wry humor. It was, of course, way too late to change the name. (I'd love to find a name that doesn't have "edit" in it.) Nevertheless, people didn't care about the name, because all they had to do was see their favorite feature — the one they'd been looking for forever — and they were hooked.

One of the things I've tried to do is to make BBEEdit appeal to a broader audience than these other text editors have done. Rather than produce a literal copy of a sample program or try to emulate Emacs, I wanted to provide a Macintosh application that took unique advantage of the Mac's capabilities, so that it would be easy to use by anyone who needed to prepare text, not just by programmers.

Despite this broad appeal, however, BBEEdit does have a few features that make it an excellent editor for Mac programmers. This is an integral part of its background.

Future Plans

BBEdit 2.1 (the first public release) was released in April 1992, and since then has become very popular, and not just among programmers. Over time, I have tried to listen to what users request. In BBEdition 2.2, most of the most-often requested features are there. Those that aren't either require serious work on the text engine or architecture, or are beyond the scope of what I believe is appropriate for this program. Some aren't here simply because I didn't have the time to do them, and will be implemented for some future release.

So what does the future hold for BBEdition? At this point, it's not completely thought out. I'd like to support files bigger than RAM. Using MultiFinder temp memory for documents is a kludge that will not stand the test of time. It should be possible to open a 40MB file with only 32K of overhead. I'd like to support live text wrapping à la TextEdit, but I want to do it without compromising the performance of the engine, and that may not be possible. A good middle ground may be to support automatic insertion of carriage returns while typing. I have a few other unique features in mind, but I'm not quite ready to tip my hand just yet.

BBEdit will probably be scriptable at some point in the future. I do not want to design and implement yet another incompatible macro language, nor do I care to support Tcl. There are all of two programs on the Mac that support it, and I think that Tcl on the Mac will go the way of a snowball in hell when AppleScript hits the streets.

BBEdit will continue to be free for a good while longer. The concept of shareware is honorable, but I'm willing to forgo the meager financial return (relative to the number of people who actually use the program) to foster wider acceptance of BBEdition in particular, and free software in general.

Miscellaneous Attributions

The original impetus for BBEdition is owed to the guys at THINK, particularly (but by no means limited to) Mike Rockhold, and Philip Borenstein, who would always lead off sentences with “How hard would it be to...”, “It would be nice if...”, and my favorite, “Why can’t you make it...?”. Special mention goes to Darrell Leblanc, who would hear “Try this, it should work now” at least once a day. Michael Kahl had much to offer from his experience in doing THINK C’s editor. He also had some special requirements, so there are still features in BBEdition which are keyed to his Chooser name. Jon Hueras was invaluable in helping out with the text engine. He and Meredith Lesly were the original implementors of the Capps package.

After I quit Symantec, I got some of the guys at GCC Technologies hooked on BBEdition. Lee Doron, Mike Conley, Mike Fryar, and Ken Hancock were merciless testers and users. Robert Munafo dumped MPW for BBEdition, bless his heart. Eric Broadbent still remains an unbeliever. It’s people like him who said Columbus was crazy. Write him a letter and tell him how much you love BBEdition, and maybe he’ll cave in. Mike Fryar and Lee Doron also reviewed the manual.

The beta sites have been a big help since before 2.1 was released. Dan Morrow, Neal Trautman, and Jamie McCarthy have been there for the long haul, and have helped make BBEdition a better product by being endlessly nit-picking and retentive.

My long-time friend and co-conspirator Patrick Woolsey has made contributions above and beyond for all of my work (not just BBEdition). Besides being my main sounding board and an overall voice of reason, he’s also been an excellent tester.

Leonard Rosenthof of Aladdin Systems contributed some dialog code and has graciously provided other technical assistance. David Schargel has likewise made BBEdition possible in its present compressed form.

Stephan Somogyi endured many late-afternoon calls from in traffic so that I could moan about one thing or another.

Aaron Hyde of Apple Computer was kind enough to send me an XTND Developer’s kit after I inquired about file translation.

Andria Nicolazzo reviewed the manual, put up with the late nights, and actually said “Yes” when I asked her to marry me.