A Dramatically Enhanced Development Environment

By William Baker

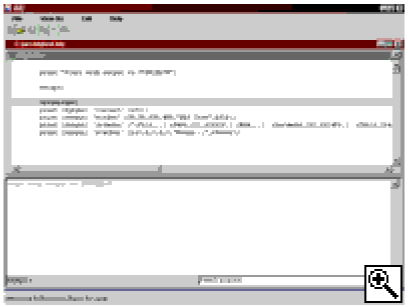
William is now a Senior Software Test Engineer in Quality Assurance and has been developing software for 14 years.

As a BBx® developer, I've had a love-hate relationship with the development environment. I've always loved the interpreted environment and the flexibility that interpreters give us over compilers. On the other side of the computer screen, the limitations of the READY> prompt have bothered me more and more over the years as other development environments have gotten more sophisticated, and as I've gotten used to Windows editors.

BBj™ will feature dramatic improvements to the development environment that give programmers modern development features while retaining all the strengths of the interpreted environment. The important new features that will make the difference are a more workable interface, breakpoints and variable watch capability.

Improved Interface

A problem with the current console mode is that screen output, program listing and console commands have to share the same screen. This makes it difficult to debug a user interface because the interface is erased by the debugging process. As you can see in the prototype screen shot on the next page, the BBj interface is designed to take care of this problem. All program listing and editing is done in the top window, program output goes to the bottom window and a vestigial READY> prompt has been relegated to a text field at the bottom. Console commands are entered in this text field and executed immediately.



BBj development environment with a program in the top window and screen output in the bottom window. Console commands are entered in the text field at bottom.

Splitting the screen into three parts like this will lead to more efficient debugging of character interfaces, and you will gain the productivity benefits of a graphical user interface. One of these benefits is a multiple-document interface: you will be able to run and edit more than one program at a time without starting multiple interpreter sessions. You will also be able to resize forms inside the window.

Console mode will still be available on terminals that don't support GUI; all output will go to one screen as it does now in console mode.

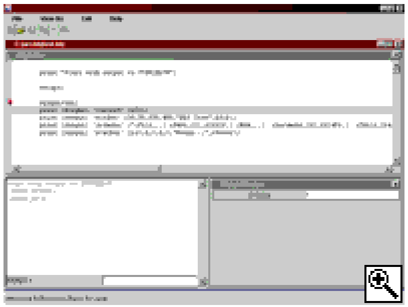
Breakpoints

A big advantage of interpreted languages is how easy it is to break out of an executing program and look around. The ESCAPE verb in BBx is very handy, but debugging will be even nicer in BBj because you can set actual breakpoints in the programming environment without modifying code. The breakpoints go away when the interpreter is closed, so you won't have the embarrassment of leaving behind ESCAPE statements by accident and getting support calls on them.

There are two ways to set breakpoints. The graphical way is to click in the editing window on the left margin of the line where you want to break. This causes a highlight indicating the location of the breakpoint to appear in the left margin. The other way is to type a command in the command field. Breakpoints are removed by clicking on the breakpoint highlights in the left margin, or they can be cleared with console commands.

Variable Watch

I think the most exciting new feature in the BBj development environment is the variable watch feature. To activate a watch, you type .WATCH and a variable name in the commands field. If the specified variable has been initialized, a watch window appears with the variable name and current value. Each time the value changes, the new value appears in the window. In the lower screen shot, you can see that two variables are being watched. To remove a variable from the watch, you type .UNWATCH and the variable name.



A watch window can be opened to display one or more variables. The new value displays each time one of the watched variables changes.

The hardest problems to debug frequently occur when an unexpected value turns up. Tracking down and checking every possible variable assignment can waste a lot of time and leave programs riddled with ESCAPE and PRINT statements. With variable watch and a few strategically placed breakpoints, it is generally a lot easier to narrow down the problem area.

These features of the new BBj development environment will make working in console mode much more pleasant and productive for any BBx developer. You'll be able to see the prototype BBj development environment and try it out at TechCon99 in September.