May, 1999

Advisor Answers

Running the Most Recent Code

Visual FoxPro 6.0/5.0/3.0 and FoxPro 2.x

Q: While working in Visual FoxPro 5.0, a problem arose in which code was edited, saved and then run, but FoxPro appears to have compiled an older version of the code and not the most recent one. Subsequent edits and saves yielded the same result, that is, the same older version of the code was run rather than any of the newer ones. When the Debugger was used to open the program, the Trace window contained the message "Source is out of Date". What's going on here?

—Chris Keller (via Advisor.COM)

A: While you haven't said so, I'll guess that you're either working with an .APP or .EXE rather than running individual programs, or that you've SET DEVELOPMENT OFF (probably in your main program) and are editing in a separate VFP session or with an external editor. The behavior you're seeing can occur in all of those situations.

One of the ways that FoxPro (all versions back as far as I can remember) speeds things up is by keeping programs you've run in memory, so that they're available more quickly if you run them again. Most of the time, that works out very well. However, in certain situations, it can drive you nuts.

Normally, when you run a program, then modify and save it, and run it again, FoxPro notices that the program has changed, recompiles it and runs the new version. However, in the situations above, for one reason or another, it doesn't work that way.

The first case is the easiest to understand. When you build an .APP or .EXE, all the code for the programs in the project is compiled into a single file (the .APP or .EXE). Changing an individual program doesn't affect what's in the compiled application. You have to rebuild the .APP or .EXE for the changes to take effect.

The other situation is a little more complicated. When DEVELOPMENT is ON or you're editing and running programs in the same FoxPro session, FoxPro always checks the date and time of a program and recompiles it if it's newer than the compiled version. But with DEVELOPMENT OFF and editing outside the FoxPro session where you're running, that check is omitted. Here's a simple way to see the effects of this setting.

Create a program, Test.PRG, that contains:

```
WAIT WINDOW "This is a test"
```

Save and run it. So far, so good. Now, open another FoxPro session. In that session, modify Test.PRG and change the line to:

```
WAIT WINDOW "This is still a test"
```

Save (but don't compile) the program and go back to the original session. Run the program there. Again, all is well.

Now, SET DEVELOPMENT OFF in the session where you ran the program. Go back to the second copy of FoxPro and change the code to:

```
WAIT WINDOW "Now we're in trouble"
```

Again, save without compiling. Go back to the first instance and run the program again. Notice that it runs the "This is still a test" version, not the latest version. SET DEVELOPMENT ON and run the program again and all is well.

As for the Trace window, its behavior makes sense although the exact message it gives you is misleading. The "Source is out of date" message appears whenever the original program and the compiled version are out of synch (as well as when the Debugger can't find the source code, often because the directory it's in isn't in FoxPro's search path). More often than not, they're out of synch because the code was changed, but not recompiled, so the message reflects that point of view. In this case, of course, you have the opposite situation: the source is newer than the compiled code. Nonetheless, the Debugger can't find the code that matches the compiled code, so it complains.

What's the solution? If you're working with an .APP or .EXE, rebuild it before running again. If you're using an outside editor, make sure that DEVELOPMENT is ON during development.

Incidentally, Help says CLEAR PROGRAM should solve this problem. It didn't in my tests. The older version was still run. That actually makes sense to me because CLEAR PROGRAM's job is to clear the cache of previously run programs. But that doesn't solve the deeper problem here of the new version not being compiled.

—Tamar