

# Using Source Debugger - A Technical Tip

submitted by [Dave Schnee](#), Barsa Consulting, LLC

This "tip-du-jour" is about using the source debugger in an environment typical of many installations. The problem I often find is that, in order to debug a program, the source debugger needs to be loaded FIRST and then IT has to call the program or execute a command. When the program you have named (or the command) completes, the debugging session ends.

This is fine for a simple situation where you start from a command line but not so fine where there is considerable setup needed (e.g., a library list, a password, 3 levels of menus, a complex call with multiple overrides, parameters and OPNQRYP usage's, etc.). It would be ever so much easier to be able, from your usual menuing system, to "get into debug" for ANY program from a simple command while you are already IN your environment (like you used to be able to do with the old, non-source, debugger as long as you could execute a command).

Well, try this one ... all it requires is that your environment allows you (as the programmer/debugger) to get to a command line while in your environment:

Create a "do nothing" program in a library you can find easily, like QGPL, and which you can call with no ill effect. The one I use is a CL program like this:

```
0.....1.....2.....3.....4.....5.....6.....7.....8
1234567890123456789012345678901234567890123456789012345678901234567890

IEFBR14:          PGM
                  RETURN
                  ENDPGM
```

(A trivia question for you old-timers --- what operating system, and what decade, did THAT program name come from?).

Start your usual programming day of testing and debugging by calling your menuing/entry system with the source debugger and naming the "do nothing" program as the object being debugged. For example:

```
0.....1.....2.....3.....4.....5.....6.....7.....8
1234567890123456789012345678901234567890123456789012345678901234567890

STRISDB          PGM(QGPL/IEFBR14) UPDPROD(*YES) INVPGM(*CMD) +
                  CMD(CALL STARTMEUP PARM('*PGMR E' 4567))
```

You can, of course, create a program that will execute this command for you if you're as lazy as I am).

The effect will be to start the source debugger and call your menuing system. Later in the day, when you need to debug someone else's code (you never need to fix YOURS, do you?) get to your command line (however you do it in your environment) and call program IEFBR14. The source debugger will start for this "do nothing" program and wait for your instructions. Press 'F14' to edit the list of programs being debugged, add the one you REALLY need to the list, press 'F12' to return to your "do nothing" program, press 'F17' to continue (and end) the "do nothing" program and you're READY. When your menuing system calls the program you have added, the source debugger will wake up, change the default program to the one just called, and away you go.

In addition, you can always, from your trusty command line, use a few of the "old" debugging commands such as 'DSPDBG' and 'DSPPGMVAR' just like you used to if you're not at a breakpoint already (in which case you have better ways to do these functions). Enjoy.

Dave Schnee.