Resetting Your Environment After a Program Crash

At some point in the development cycle, your system is going to crash - and it will take the environment with it. Your carefully customized development environment, with special hot-keys and macros, SET settings, paths, colors, and a whole host of other options that you've tuned specifically for your system are going to go up in smoke.

The more you've done to customize, the longer it will take to recover. Most every developer finds themself typing in a half dozen commands to reset the environment - and even then, misses a few, or gets tired and leaves the rest.

Furthermore, depending on what stage in the program you were in, some of the settings you started out with may have been changed. For instance, you may normally keep the setting of EXACT set to OFF. When you need to EXACT to be set to ON, you'll do so in your program, issue the commands you need, and then change it back.

Let's suppose your program crashed before EXACT was set back to OFF. The next time you run your program, EXACT is still ON, and, as the saying goes, unexpected results may occur.

Thus, one tool you should have in your development directory should be a program that resets your environment back to the way you want it.

Here is a sample program that you can use as a starting point. You'll probably find commands that you want to pull out, and other commands that you need to add because of your environment.

I call my program "Z" - it only takes five keystrokes (or two mouse clicks) to reset FoxPro's environment back to the state I want it.

```
* Z.PRG
*
* version for 2.5/D/W
*
wait wind nowa "Resetting developer environment"

*
* clean up memory and screen
*
clear program()
set library to
set procedure to
clear read all
clear all
close all
clear menu
clear window
clear

*
* reset keyboard
*
on key
on key label rightmouse keyboard "{ENTER}"
on key label ALT-F10 quit
on key label F2 do _WhilHlp
on key label F3 do _ToDo
restore macro from sys(2004)+"DEFAULT.FKY"
```

```
on error
on escape
set help to sys(2004)+"FOXHELP"

*
* environment specific settings
*
do case
case _DOS
 set clock on
 set clock to 0,72
 set color set to no_shad_1
 set display to vga5
 set hour to 24
case _WINDOWS
 set clock status
 set status bar on
endc
set resource to g:\foxtemp\foxuser

*
* specific SETs
*
set century on
set debug on
set delete off
set escape on
set exact off
set notify on
set safety off
set step off
set sysmenu to defa
set sysmenu on

*
* the crash may have goofed up my path
*
set path to \common25

*
* resets default directory
*
set default to ..\source

*
* clean any remaining garbage from the screen
*
clear

return .t.
```

There are two items that may need a bit of explaining. In the "reset keyboard" section, the SYS(2004) function is used twice. This function returns the location (including drive) from which FoxPro was started. Thus, this same program can be run with every version of FoxPro, and still refer to files in a specific FoxPro directory.

Second, in the "reset default directory", the "..\source" syntax may confuse some people. I keep my programs and data files in two directories on the same level, and a program crash will normally happen once I've switched into the data directory. This command just switches back to the program directory (called SOURCE).

This type of program has three benefits.

The obvious benefit is the time savings accrued by not typing the same commands in over and over again. A more subtle benefit is the time savings by not having to hunt down those insidious bugs that just "appeared" and are a result of not having set the environment back to exactly the way you wanted it.

The third benefit is improved quality. By ensuring a consistent starting point for your programs, you decrease the chances of random anomolies creeping into your system.

<EOF>