

Do custom replacements with GoFish

The Search tool lets you write code to replace matches when you need more than a simple substitution.

Tamar E. Granor, Ph.D.

I've been using GoFish (available on VFPX) for searching within VFP code for a couple of years, but I've rarely used it to replace code. When a tricky replacement task came along, GoFish had the chops to let me do it quickly and easily.

A while back, I inherited a project that was originally written in FoxPro 1.0 around 1990 and hadn't been modified in more than a decade. The project included a procedure file (named Library.PRG), but a quick check indicated that only a few of the contained procedures were actually in use.

So I decided to break the procedure file up into individual PRGs. (Though it's not the focus of this article, in my view, procedure files are obsolete and should be avoided.) Having done so, I needed to do something about all the SET PROCEDURE lines that referenced that file. Fortunately, none of the SET PROCEDURE commands referenced multiple files.

GoFish told me that the line appeared 83 times in the project. I didn't want to have to double-click each instance, comment it out and save it.

Then I remembered that GoFish can do some fairly sophisticated replacements. I dug in and rediscovered Advanced Replace mode, which lets you write code to do the actual replacement. To use it, you check the Replace Mode checkbox and then choose Advanced Replace, as in [Figure 1](#). Click Select UDF to point to the routine to run.

The replacement routine receives a single parameter, the line to be replaced; it's called separately for each line to be replaced.

It didn't take long to see that the code I needed was pretty simple; it's shown in [Listing 1](#).

Listing 1. This code comments out each matched line that starts with "SET PROC" and includes the string "LIBRARY".

```
LPARAMETERS tcOldLine

LOCAL cReturn, cOldLine, cIndent, nPrefixPos

cOldLine = UPPER(ALLTRIM(m.tcOldLine, ;
    ' ', CHR(9)))
IF LEFT(m.cOldLine, 8) = "SET PROC" AND ;
    "LIBRARY" $ m.cOldLine
    cReturn = "!*" + m.tcOldLine
ELSE
    cReturn = m.tcOldLine
ENDIF

RETURN m.cReturn
```

The code first removes any leading or trailing spaces or tabs from the line (taking advantage of the VFP 9 improvement in the TRIM functions that lets you specify characters to be trimmed) and converts it to uppercase for easier matching.

Then it checks whether the modified line begins with the string "SET PROC" and includes the string "LIBRARY". (I couldn't check that the line ended with "LIBRARY" because some lines referred to "LIBRARY.PRG".) If the line passes that test, the return string is set to "!" plus the original text. If the line doesn't pass the test, the original line is returned.

Advanced Replace replaces the original line with the return value of the specified function. What's very cool is that it actually previews the

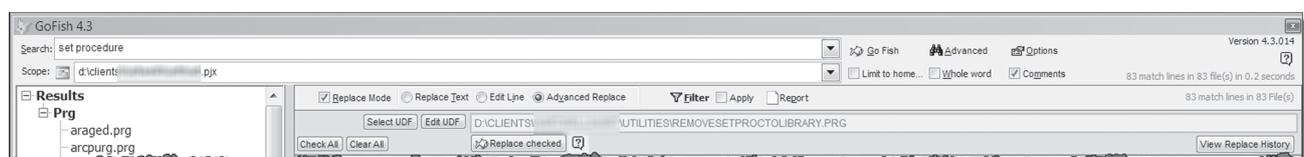


Figure 1. GoFish's Advanced Replace mode lets you write code to do the actual replacement for each line found.

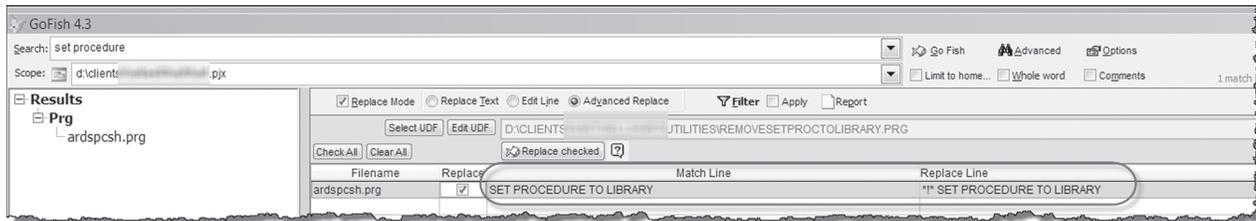


Figure 2. Once you specify the function and check the Replace checkbox for an item, GoFish shows you the result of applying the function.

replacement. Once you've specified the function, when you check the Replace checkbox for an item, the new line appears, as in **Figure 2**.

Once you're satisfied that your replacement function does what you want, click the Replace Checked button to perform the actual replacements. Not surprisingly, since you could do a lot of damage with this option, you get a warning before replacements start. In addition, by default, GoFish creates a back-up for each file it changes.

I could have written code to do the whole job from finding the lines through replacing them. But that would surely have taken longer than simply replacing each line manually. GoFish let me write only the custom part, which was quick, easy and probably more accurately than doing the task manually.

Author Profile

Tamar E. Granor, Ph.D. is the owner of Tomorrow's Solutions, LLC. She has developed and enhanced numerous Visual FoxPro applications for businesses and other organizations. Tamar is author or co-author of a dozen books including the award winning Hacker's Guide to Visual FoxPro, Microsoft Office Automation with Visual FoxPro and Taming Visual FoxPro's SQL. Her latest collaboration is VFPX: Open Source Treasure for the VFP Developer, available at www.foxrockx.com. Her other books are available from Hentzenwerke Publishing (www.hentzenwerke.com). Tamar was a Microsoft Support Most Valuable Professional from the program's inception in 1993 until 2011. She is one of the organizers of the annual Southwest Fox conference. In 2007, Tamar received the Visual FoxPro Community Lifetime Achievement Award. You can reach her at tamar@thegranors.com or through www.tomorrowssolutionsllc.com.