March, 2002

## Advisor Answers

## Returning arrays

VFP 7

Q: I've heard people say that you can return an array from a function in VFP 7, but I can't make it work. What am I missing?

–Name withheld by request

A: Your informants are correct. VFP 7 allows you to return an array. However, the conditions for doing it mean that it's only practical when the "function" is a method and the array is a property of the same object.

The rule is that the array must be in scope after the function finishes running. So, it can't be a local or private variable. If an array already exists in the calling routine, you might as well pass it as a parameter and let the function modify it. Here's an example that shows how you can return an array, but it uses bad coding practices:

```
* Calling code
* This is an example of what NOT to do
PUBLIC aResult[3]
LOCAL aHoldResult[3]

aHoldResult = MyFunc()

RETURN

* Here's the function
FUNCTION MyFunc

* Do something with the array
aResult[1] = 7
aResult[2] = 12
aResult[3] = 17

RETURN @aResult
```

It's easy to see that the function depends on the existence of the aResult array and that there's really nothing gained in this situation (and a fair amount lost in readability and maintainability). The example does show one important item, though – to return an array, you precede it with the "@" symbol in the RETURN statement.

However, when you're dealing with a class definition, the ability to return an array is pretty cool. Consider the problem of taking color values apart into their red, green and blue (RGB) components. The FoxTools library contains a function RGBComp that does this, but in addition to the color value, you have to pass three variables by reference to hold the results. With the new array return capability, we can put the values into an array and return them together.

Here's the class code, available in Colors.PRG on this month's Professional Resource CD (which also contains code to demonstrate the technique):

```
DEFINE CLASS Colors AS Custom
* Color handling code
DIMENSION aRGB[3]

FUNCTION RGBComp(nColor) AS Array
* RGBComp
* Returns the Red, Green and Blue Components
* of a color in an array

This.aRGB[1] = -1
This.aRGB[2] = -1
This.aRGB[3] = -1

IF VARTYPE(nColor)="N"
   This.aRGB[3] = INT(nColor/(256^2))
   nColor = MOD(nColor,(256^2))
   This.aRGB[2] = INT(nColor/256)
   This.aRGB[1] = MOD(nColor,256)
ENDIF

RETURN @This.aRGB

ENDDEFINE
```

You might call the method like this:

```
LOCAL oColors, nColor, aColors[3]

nColor = 8388736
oColors = NewObject("Colors","Colors.PRG")
aColors = oColors.RGBComp(nColor)

? "The original color is " + TRANSFORM( nColor )
? "Red = " + TRANSFORM(aColors[1])
? "Green = " + TRANSFORM(aColors[2])
? "Blue = " + TRANSFORM(aColors[3])

RETURN
```

In this case, because the array aRGB is a property of the class, it continues to exist when the method is finished, and thus the values in the array can be copied into the aColors array in the calling program. There's one other restriction on the returned arrays. The array you assign the result to must be a variable, not a property of another object.

The VFP Help indicates that this was made primarily for building COM servers, but I think we'll be finding lots of uses for it in conventional VFP code, as well.

–Tamar