

March, 1997

Advisor Answers

Visual FoxPro

Q: Do you know a way I can reference objects (like THISFORM.<some member> or GRID1.COLUMN1.<something>) within .PRG files. Apparently, even if all the objects are defined they cannot be referenced except in method code. I have a routine I want to call from two different forms. I guess I can stick it in the application class. Is this the best way? Is there another?

–Mark Perrone (via CompuServe)

A: Depending on the nature of the routine and the forms, there are two answers to this question. First, you need to consider whether the two forms have enough in common that you should create a form class containing the routine as a custom method, then derive each of these forms from that form class. For example, if both forms are data entry forms and the routine is for validating data of a certain type, you might create a data entry form class containing that method (and any others related to data entry), then base the two forms on that class.

If the routine in question really doesn't seem appropriate as a form method or is perhaps a method of another class, you can pass a parameter to it to tell it which form or object it's operating on. The key point here is that objects can be passed as parameters just like other data. When you call the routine you pass THIS or THISFORM to it. For example,

```
DO MyRoutine WITH THIS
```

passes a reference to the current object to MyRoutine while

```
=MyFunction(THISFORM)
```

passes a reference to the current form to MyFunction.

In the routine, provide a parameter to hold the object reference, like this:

```
LPARAMETERS oTarget
```

Then, in the code, you can simply reference the object using the parameter.

Here's a routine that changes the font of whatever you pass it (if it has font properties) to bold italics:

```
PROCEDURE ShowOff
```

```
* Change font of any object to bold italics
```

```
LPARAMETER oTarget
```

```
* Did we get a good object reference?
```

```
IF TYPE("oTarget")="O" AND NOT ISNULL(oTarget)
```

```
    * Does it have FontBold?
```

```
IF TYPE("oTarget.FontBold")<>"U"  
    oTarget.FontBold = .T.  
ENDIF  
  
* Does it have FontItalic?  
IF TYPE("oTarget.FontItalic")<>"U"  
    oTarget.FontItalic = .T.  
ENDIF  
ENDIF  
  
RETURN
```

Now any object can call the routine like this:

```
=ShowOff(THIS)
```

and get its own properties changed.

One final point. One of the examples you gave - GRID1.COLUMN1.<something> - isn't going to work, even in the form that contains the grid. Visual FoxPro always requires a complete path from the outermost container (either a form or formset) to the object of interest. You'd need something like THISFORM.GRID1.COLUMN1.<something>.

The technique of passing an object reference as a parameter provides tremendous flexibility. Keep this approach in mind whenever you have a routine that needs to refer to unknown objects.

-Tamar