

June, 2001

Advisor Answers

Adding Objects at Runtime

Visual FoxPro 7.0/6.0/5.0/3.0

Q: How can I create text boxes on the fly? I have an empty form and want to create a text box object for every record in a table. (There will be no more than 11 records.) How do I get VFP to create these objects when the form opens?

–Patrick Sturge (via Advisor.COM)

A: Adding objects to a form at runtime is no problem. Use the form's AddObject method. In your situation, you might do something like this in the form's Init method:

```
LOCAL nRecNo, cTxtName, oTextBox

SELECT YourTable

nRecNo = 0
SCAN
    * Create a name for the new textbox
    nRecNo = nRecNo + 1
    cTxtName = "txtRecord" + PADL(nRecNo,2,"0")

    * Add it
    This.AddObject(cTxtName, "Textbox")
    oTextBox = GetPEM(This, cTxtName)
    WITH oTextBox
        .Value = YourField
        * Position it - add these appropriately
        .Top = (nRecNo-1) * (.Height + 8)
        .Left = 14
        .Width = 120

        * Make it visible, since added objects are invisible by default
        .Visible = .T.
    ENDWITH
ENDSCAN

* Resize the form
IF RECCOUNT() > 0
    This.Height = nRecNo * (oTextBox.Height + 8) + 14
ENDIF
```

The code loops through the table, using AddObject to add one textbox for each record. In an application, you'd probably want to use a

textbox subclass in AddObject rather than the base textbox class. In that case, you might use the NewObject method, introduced in VFP 6, instead. It allows you to specify both the class name and class library. However, NewObject is slower than AddObject, so in a loop like this one, the difference between that approach and issuing SET CLASSLIB once and then using AddObject may be unacceptable. You'll need to test in your environment.

A unique name for each textbox is generated as "txtRecord" plus a two-digit value that is incremented for each record.

Once the textbox has been added, it's given a value based on the data in the record. Then it's moved from the default location (0,0). The code that sets the Top property puts each textbox 8 pixels below the bottom of the one above. The left edge is arbitrarily set to 14 and the width to 120 pixels. The spacing was chosen to follow Windows guidelines for 10-point Arial, but you can either choose hard-coded values that work in your situation or do some computations to position and size the textboxes as you need them. Finally, the form is resized so that all the textboxes are displayed.

Having done this, I suspect it isn't really the right solution for your problem. These textboxes aren't actually tied to the record data. While a user can change the displayed values, nothing is done with the new data.

You haven't indicated why you want to show each record, but generally there are two reasons. Either you want to allow users to edit the data or you want to allow them to choose a record.

In the first case, a grid is a better choice than individual textboxes. It lets users see and edit data from multiple records at once. Set the grid's RecordSource property to the alias for your table and VFP does the rest. If you want to limit the grid to only certain fields, set its ColumnCount property to the appropriate number, then set the ControlSource of each column to the name of the field you want to appear in that column.

If the goal is to allow users to choose a record, a listbox is effective. Again assuming that you don't want all fields displayed, set RowSourceType to 6-Fields, then set RowSource like this:

```
Alias.Field1, Field2, etc.
```

substituting the Alias for your table and the names of the fields you want displayed. Set ColumnCount to the number of fields to display

and set ColumnWidths so that each field has enough room. ColumnWidths takes a comma-separated list of numbers. You can set ColumnLines to indicate whether or not line should be drawn between columns.

With a short list like yours, you can also use a combobox to display data from each record. Set it up the same way as the listbox.

The key point is to consider your goal in displaying this data and to choose the appropriate control.

-Tamar