

November, 2004

Advisor Answers

Anchoring controls on forms

VFP 9

Q: I'd like to let users resize the forms in my application, but when they do so, the controls on the forms don't adjust to fill the space. Is there a way to make them adjust?

A: Prior to VFP 9, adjusting the contents of a form when it was resized was a lot of work. You had to store information when the form opened and then do some geometry on the resize. (My article "Let Users Control Fonts" in the November, 2002 issue mentions one approach to this problem.)

VFP 9 makes the solution as easy as setting a single property for each control on the form. The new Anchor property lets you specify the behavior of a control when its container is resized. Because the possible values for the property are complex, VFP 9 even includes a tool to make it easier to specify the right value.

Adjusting controls on a form is one of those problems that initially seems fairly easy, but gets more complicated the more you look at it. It's not enough to simply resize each control the same amount as or proportionally to the change in the form, because that might result in controls overlapping. Any working solution must also reposition controls.

Even if you both resize and reposition the controls, your results may not be very attractive. It turns out that the behavior you want varies with both the control and its position on the form. For example, when a form gets both wider and taller, a textbox should probably get wider, but an editbox should increase in both directions. If the textbox is above the editbox, it probably should stay where it is, but if it's below the editbox, it needs to move down.

VFP 9's Anchor property takes these considerations into effect and lets you determine the exact behavior of each control. For each border of a control's container, you can specify that the control is anchored to that border absolutely (so that the distance between the border and the control never changes) or relatively (so that the distance between the border and the control is computed based on their original relationship

and the changes to the form) or that the control isn't anchored to that border at all. You can also take an alternative approach for any given control, and anchor its center either horizontally or vertically.

The combination of values you specify determines whether the control moves, changes size or both when the container's size changes. For example, a control anchored to both the top and bottom gets taller or shorter as its container does, though the amount it changes depends on whether it's anchored absolutely or relatively. A control anchored to the bottom and to the right moves with the southeast corner of the container. (This is often a good setting for buttons like OK and Cancel that should be kept near the bottom right edge of a form.)

Not surprisingly, Anchor is an additive property; you specify its value by adding together the values for the settings you want. The VFP Help topic for Anchor includes a table showing each of the possible values. For example, 4 indicates anchoring absolutely to the bottom border, while 8 anchors absolutely to the right border. So, to specify absolute anchoring to both the bottom and right borders (as for an OK button), set Anchor to 12.

Figure 1 shows a simple form designed to be resized. The Name and City textboxes adjust their width as the form's width changes. The Address editbox changes size both horizontally and vertically as the form changes. The Save and Cancel buttons keep themselves centered horizontally, but move vertically with the bottom of the form. The City textbox, the State combo and their respective labels move up and down to stay out of the way of the Address editbox. Figure 2 shows the same form after it's been resized. Table 1 shows the Anchor settings for each control on the form. The form is included on this month's Professional Resource CD as Anchoring.SCX.



Figure 1. Prepared for resizing—The Anchor property for each control on this form has been set to keep it attractive as the form size changes.



Figure 2. Resized—After making the form wider and taller, the controls are still organized in a usable way.

Table 1. Anchor settings—These settings, used in the example form, adjust controls as you'd expect when you resize the form.

Control(s)	Anchor	Meaning
lblName (Name label)	0	Anchor absolutely to top left corner.

Control(s)	Anchor	Meaning
lblAddress (Address label)		Don't move or resize.
txtName (Name textbox)	10	Anchor absolutely to left and right sides, so resize horizontally.
edtAddress (Address editbox)	15	Anchor absolutely to all four borders. Resize both horizontally and vertically.
lblCity (City label) lblState (State label) cboState (State combobox)	4	Anchor absolutely to bottom border. Move vertically.
txtCity (City label)	14	Anchor absolutely to left, right and bottom borders. Resize horizontally and move vertically.
cmdSave (Save button)	132	Anchor absolutely to bottom border and relatively to right border. Move horizontally to keep relative to original position. Move vertically.
cmdCancel (Save button)	36	Anchor absolutely to bottom border and relatively to left border. Move horizontally to keep relative to original position. Move vertically.

Figuring out what value you want for each control can be tricky. To make it easier, VFP 9 comes with a tool called the Anchor Editor (shown in Figure 4). With the Anchor property selected in the Property Sheet, clicking on the ellipsis button next to the textbox where you type in values opens the Anchor Editor.

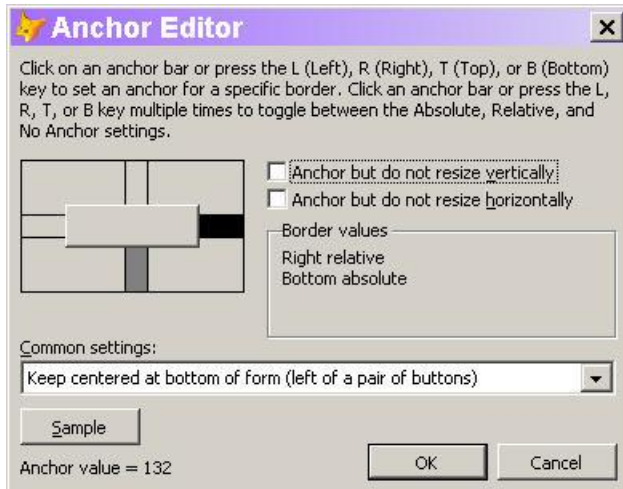


Figure 4. The Anchor Editor—This new tool makes it easier to set the Anchor property.

The Anchor Editor provides three ways to set the property. The two checkboxes offer the settings for anchoring the center of the control horizontally or vertically. The diagram lets you specify each of the four borders separately. Click between the parallel lines to change the setting for a particular border. There are three possible settings for each border (no anchoring, relative anchoring, absolute anchoring); clicking cycles among them. The dropdown combo offers common settings; the list of choices varies with the type of control. (In Figure 4, the Anchor Editor was opened for a button.)

The Anchor Editor shows your current choices in several ways. In the diagram, the area between the parallel lines uses the background color if no anchoring is specified for that border, medium gray for absolute anchoring, and black for relative anchoring. In addition, the current settings are shown in words inside the Border values box, and the numeric value for the Anchor property appears at the bottom of the form.

There's also a way to test your current settings. Click the Sample button and a form opens with your control centered. Resize the form to see the behavior of the control with the anchor settings currently specified in the Anchor Editor. The sample form is modal, so you have to close it before you can change the settings.

When you click OK to close the Anchor Editor, the specified value is stored in the Anchor property of the control.

The Anchor Editor is written in VFP 9, so if you don't like it, you can replace it with your own version. It uses a new feature of VFP 9 that

allows you to specify your own editors in the Property Sheet. Look for an article explaining this technology in a future issue.

-Tamar