Drill Down into Your Objects

Whil Hentzen

"What this country needs is a good five cent anything." Or so goes the witticism. But it's not true – there are plenty of good things around – and some of them won't even set you back five cents. Last month I showed you eView (Erik View? Extended View? Extra-special View?) This month's inexpensive but invaluable treasure is the CodeMine Object Inspector.

Ever had a form running, and wonder just what you called the text box that holds the account number? Was it txtNoAccoount? Or txtAccountNo? Or even txtAccountNumber? Object Inspector to the rescue. Call it up, click on the Enable button, and then Ctrl-Shift-Click on an object to have its properties and methods displayed in the right pane of the Object Inspector window. You can even drill down into arrays and change the values of properties on the fly.

### The basics of using the Object Inspector

The basic Object Inspector screen is shown in Figure 1. The left pane shows all of the objects currently in memory, and the right pane shows all of the properties and methods of the object highlighted in the left pane.

CT-01.TIF
*Figure 1. The CodeMine Object Inspector displays objects in the left pane.*

You don't have to do anything fancy – like actually having a form running - to get started with the Object Inspector. Just load the Object Inspector and then double click on _SCREEN to examine what's up with the _screen object. You'll get a screen like that shown in Figure 2.
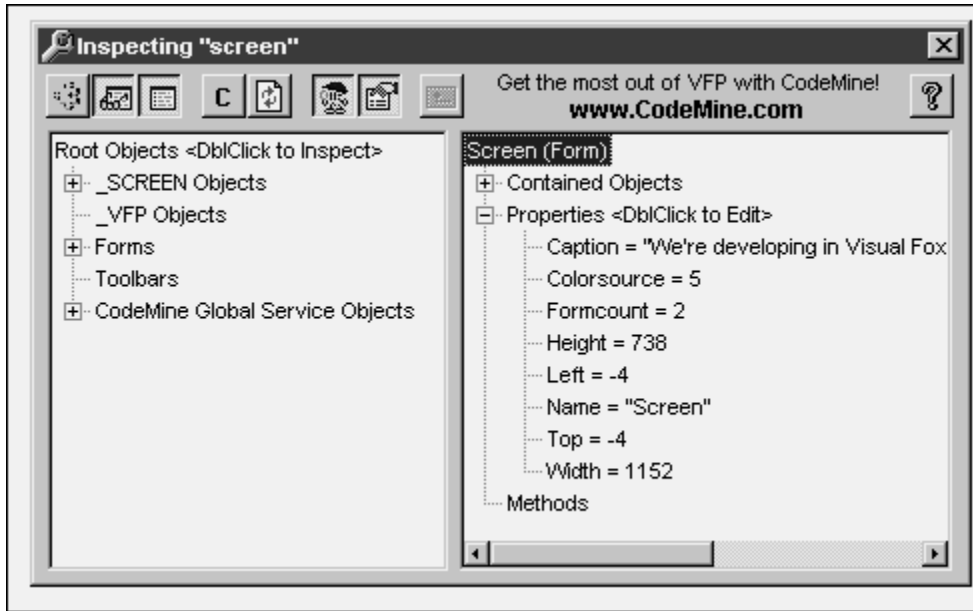
CT-04.TIF
Figure 2. The CodeMine Object Inspector displays Properties and Methods in the right pane for an object selected in the left pane.

Notice that a lot of properties seem to be missing – like FontName and FontSize, for example. By default, only non-default properties are shown in the right pane. In order to see all properties, click the second button from the right (the one with the hand pointing to the box underneath it). You'll get all properties, as shown in Figure 3.
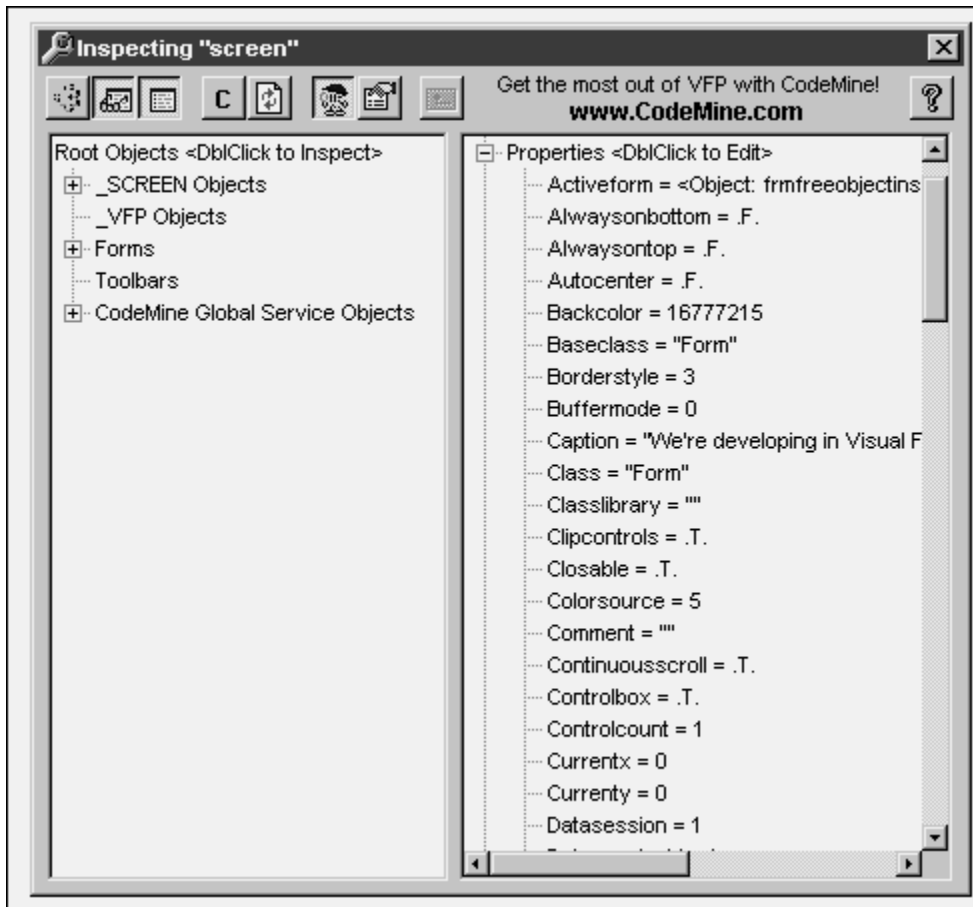
CT-05.TIF
Figure 3. The Show Default Properties Only button toggles between all properties and non-default properties.

Viewing is great – but what if you want to change something? You can do that in the Visual FoxPro Debugger, albeit with a bit of difficulty sometimes. To do so here, just double click on the property. If the property type is a logical, the value will automatically change from True to False (or vice versa.) Other property types will bring up a dialog, as shown in Figure 4, where I double-clicked on the _SCREEN's caption property.
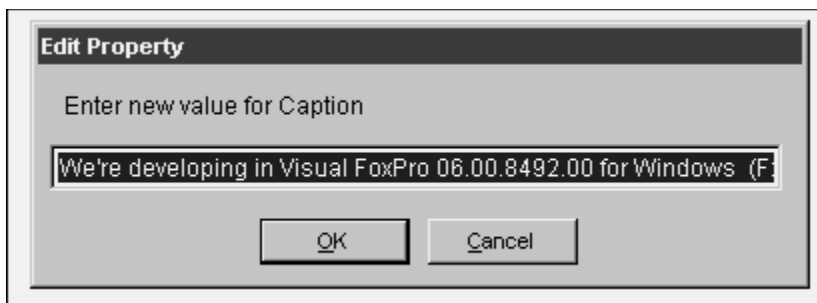

CT-06.TIF
Figure 4. Double-clicking on a properties' node in the right pane opens the Edit Property dialog that allows you to edit it.

So far, so good. But what about real forms? Figure 5 shows an actual form, running in an actual application (kids, don't try this at home without an adult!), and the CodeMine Object Inspector displaying the properties for the Original Password text box.

What I did was bring up the Object Inspector, and clicked on the sixth button (the one to the left of the pointing hand). This action enabled the Object Inspector's trapping mechanism. Then I ran my app, and brought forward the Change Password screen. I held down Ctrl and Shift, and clicked on the Original Password text box. This fires a SYS(1270) function inside the inspector, grabbing an object reference to the control under the mouse, and then the Inspector goes from there.
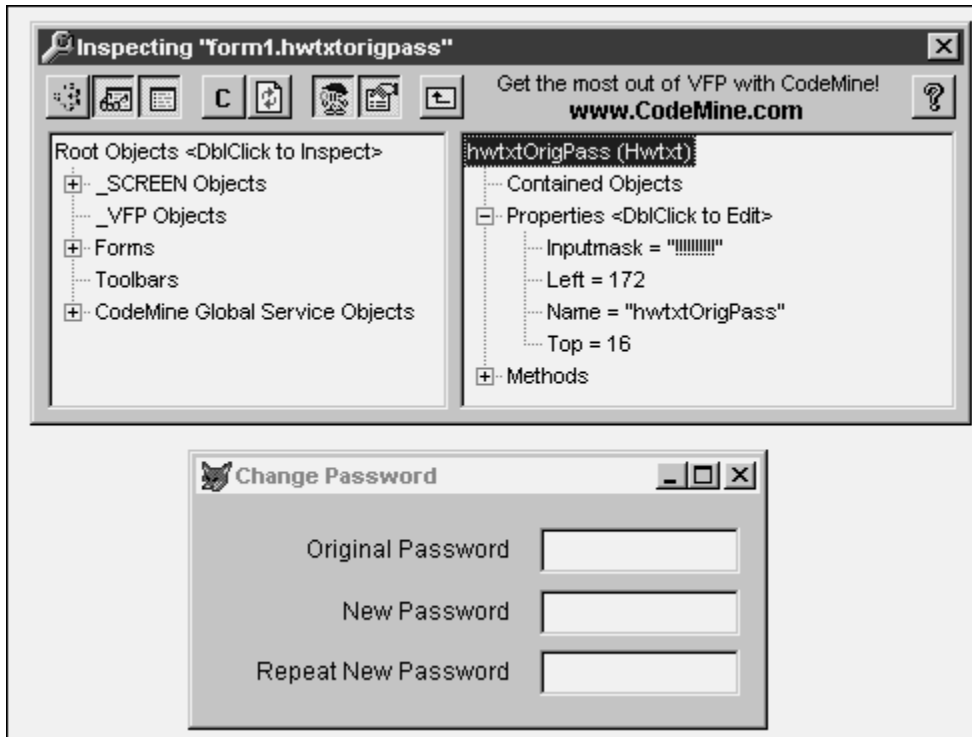
CT-02.TIF
Figure 5. Using the CodeMine Object Inspector in concert with a live form.

As you would probably suspect, modifying the value in the Object Inspector will change the corresponding attribute in the form. This can get tricky when dealing with data, for example. In Figure 6, I changed the value of the Original Password text box from being blank to "HERMAN". Then, when I clicked on the Change Password dialog, the word "HERMAN" was displayed in the form automatically.
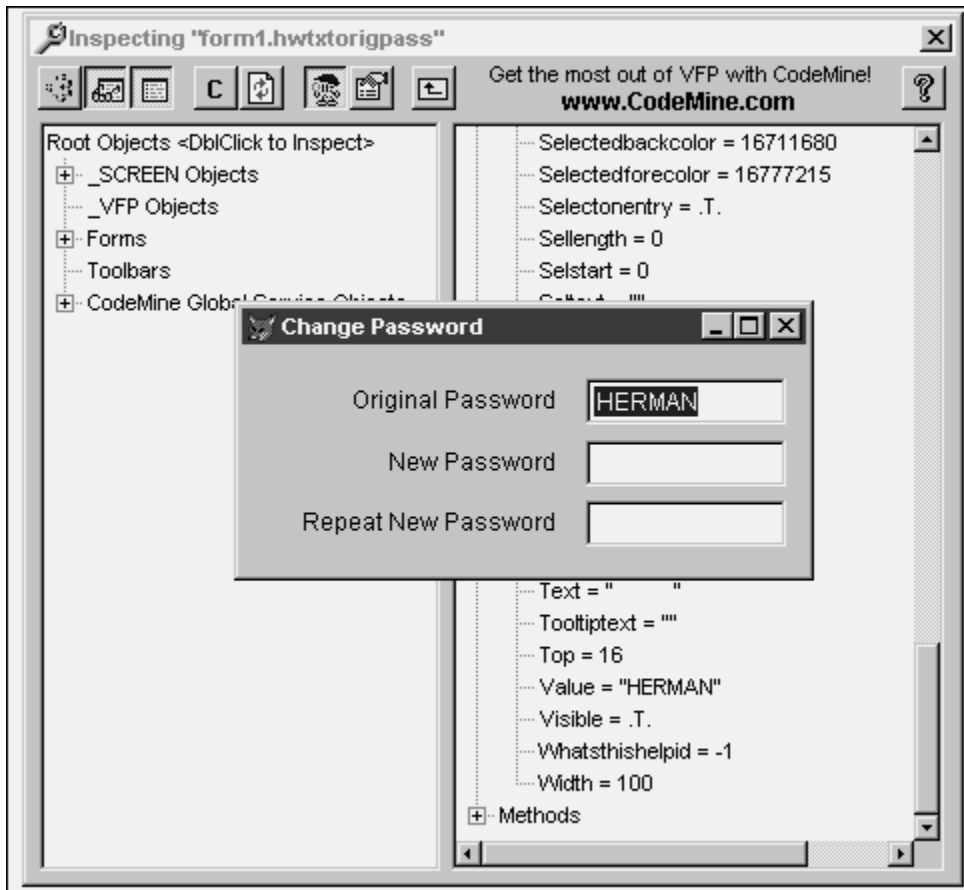
CT-07.TIF
Figure 6. You can change any property of an object – even the value property of a text box – and have it displayed in your application immediately.

It's time to discuss what the rest of the buttons in the Inspector's toolbar do. They are, from left to right:

Debugger – brings up the VFP debugger.

Data Session – opens or closes the VFP Data Session window

Command – opens or closes the VFP Command window

Clear – clears the right pane and removes any outstanding object references

Refresh – refreshes the tree view in the right pane. For example, if you change the value of a control in the form, the Object Inspector doesn't automatically pick up the new value. Clicking refresh will do so – it's sort of like a Requery() for the Object Inspector.

Enable Object Inspector – as discussed earlier, this enables the Ctrl-Shift-Click hot key that allows you to select an object to inspect

Show Default Properties Only – also as discussed earlier, this toggles between the displaying all properties and just the non-default properties. Note that if you display all properties, the non-default peorpteis are not displayed in bold like they are in the VFP Properties window.

Inspect Parent – This automatically grabs an object reference to the parent of the selected object, and displays it in the right pane.

The CodeMine Object Inspector is freeware: you may use and re-distribute it as much as long as you like as long as you don't modify it in any way from its original form. A copy of it is included with this month's Subscriber Downloads.

The toughest part of using it is remembering to do so – but once you get in the habit, you'll wonder why the developer in the cube next to you is taking so long to spelunk through their code.

Whil Hentzen is editor of FoxTalk.