

August, 2000

Advisor Answers

Using the Internet Transfer Control

Visual FoxPro 6.0/5.0

Q: I've been playing with the Microsoft Internet Transfer Control (inetctls.inet.1) and it would be great if only it worked.

If I instantiate the control directly, none of the methods work. For example, I've tried:

```
objInet = createobject("inetctls.inet.1")
?objInet.openUrl("http://www.advisor.com")
```

I would expect this to print all the HTML code for the corresponding web page to the screen; however, nothing happens. I have a similar experience with all the other methods. The properties work fine.

Using the control via an olecontrol causes VFP to crash when the object is released. The following code crashes VFP for me every time:

```
ofrm = createobject("frminet")
ofrm.inet.execute("http://www.advisor.com")
activate screen
?ofrm.inet.getheader()
* SUSPEND && crash happens on following line
ofrm.release
DEFINE CLASS frminet AS form
    Top = 0
    Left = 0
    Height = 39
    Width = 132
    DoCreate = .T.
    Caption = "Form2"
    Name = "frminet"
    ADD OBJECT inet AS olecontrol WITH ;
        Name = "inet", ;
        OleClass = "inetctls.inet.1"
ENDDDEFINE
```

Any pearls of wisdom you can offer would be greatly appreciated.

—Rhodri Evans (via Advisor.COM)

A: The Internet Transfer Control is designed to make it easy to move documents using both the HTML and FTP protocols. Like many other controls, it can be used either as an ActiveX control or directly as an

Automation server. Also like those other controls, when you use it as a server, you have to do more of the work yourself. When a server is packaged as a control, a number of default values are set for you.

In the case of the Internet Transfer Control, there's only one important property value that needs to be set before you can get it working. That's the RequestTimeout property. When you instantiate the control with CreateObject(), RequestTimeout is set to 0 (as are all of the control's other numeric properties; character properties default to the empty string). That prevents methods like OpenURL from working, since they'd then require the specified website to respond instantaneously. When you add the ActiveX Internet Transfer Control to a form, RequestTimeout defaults to 60 seconds, a far more reasonable value.

With this additional information, we can now get your first example working, as follows:

```
objInet = CreateObject("inetctl.inet.1")
objInet.RequestTimeout = 60
cHTML = objItem.OpenURL("http://www.advisor.com")
?cHTML
```

On the whole, it's often easier to use a control by putting it on a form and working with it there. Then you have the default values that the control's creator felt were appropriate as well as VFP's visual interface for working with the control.

Like a number of other controls that provide services, but don't have anything visual to offer, the Internet Transfer Control is visible at design-time, but not at runtime. To add it to a form, you must first make it available. In the Tools-Options dialog, choose the Controls page (Figure 1), then choose the ActiveX controls option button. Make sure that Microsoft Internet Transfer Control is checked. (I'm using Version 6.0 of the control.) Choose OK to make the control available in the current VFP session. To make the control available for all future sessions, choose Set as Default before choosing OK. (Be aware that this makes all the settings in Tools-Options your defaults for future sessions.)

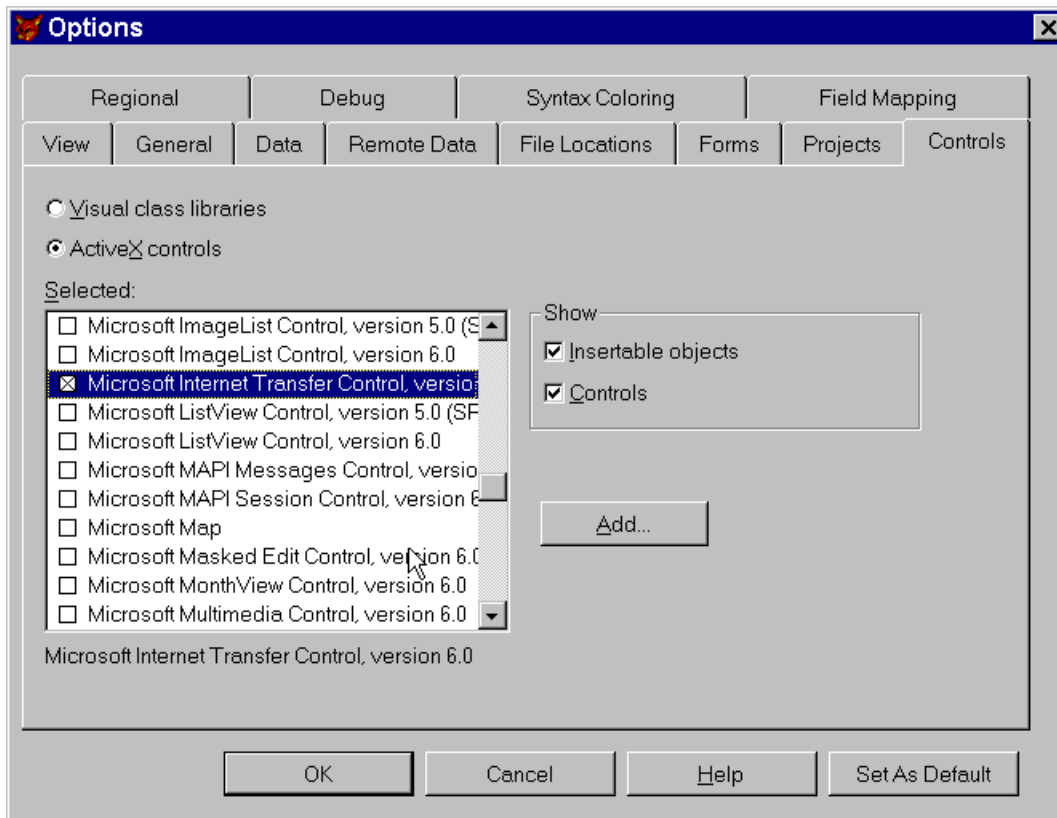


Figure 1 Making ActiveX Controls Available – Check the control you want to use in the Tools-Options dialog. To make it available from here on out, click Set as Default.

Once the control is available, open the Form Designer. Using the Form Controls toolbar, click on the View Classes button (the one that looks like a set of books). Choose ActiveX Controls and the toolbar changes to show all the ActiveX controls that have been registered either permanently or for the current session. Click on the Internet Transfer Control, then on the form to add the control to the form. (Fortunately, it takes a lot more work to describe the process than to actually do it.)

I've created a form that contains an Internet Transfer control, a textbox to enter a URL, an editbox to show the data returned, a button to start the transfer, and another editbox I'll discuss a little later. It's shown in design mode in Figure 2 and at runtime in Figure 3. It's also included on this month's PRD.

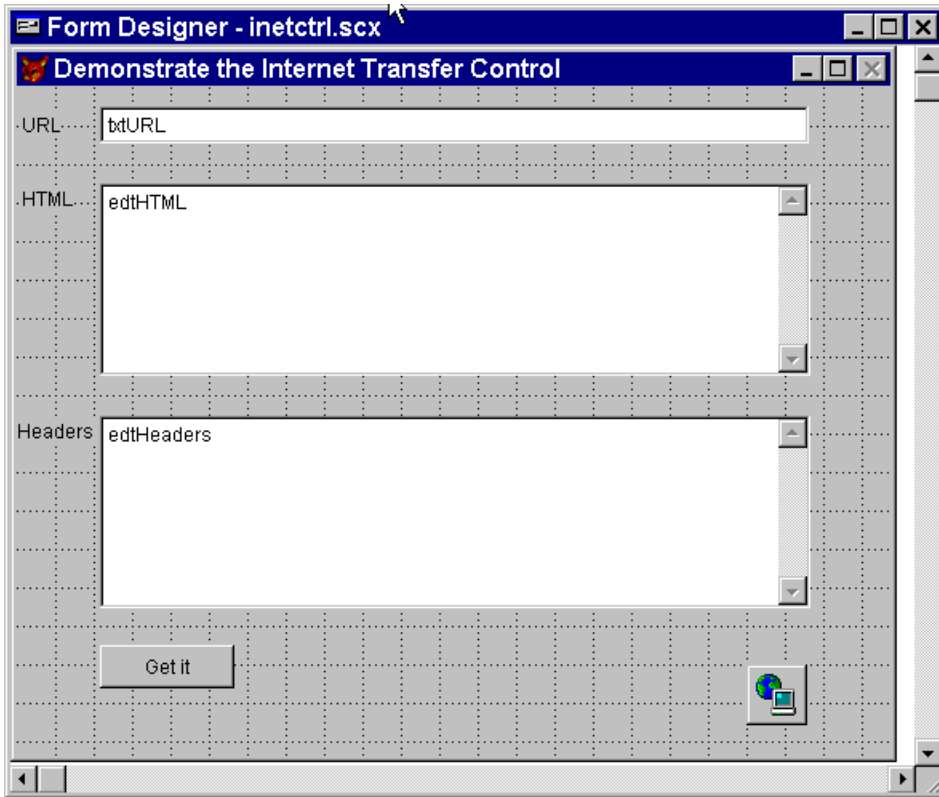


Figure 2 Adding the Internet Transfer Control—At design time, the control is visible. In this form, it's in the lower right corner.

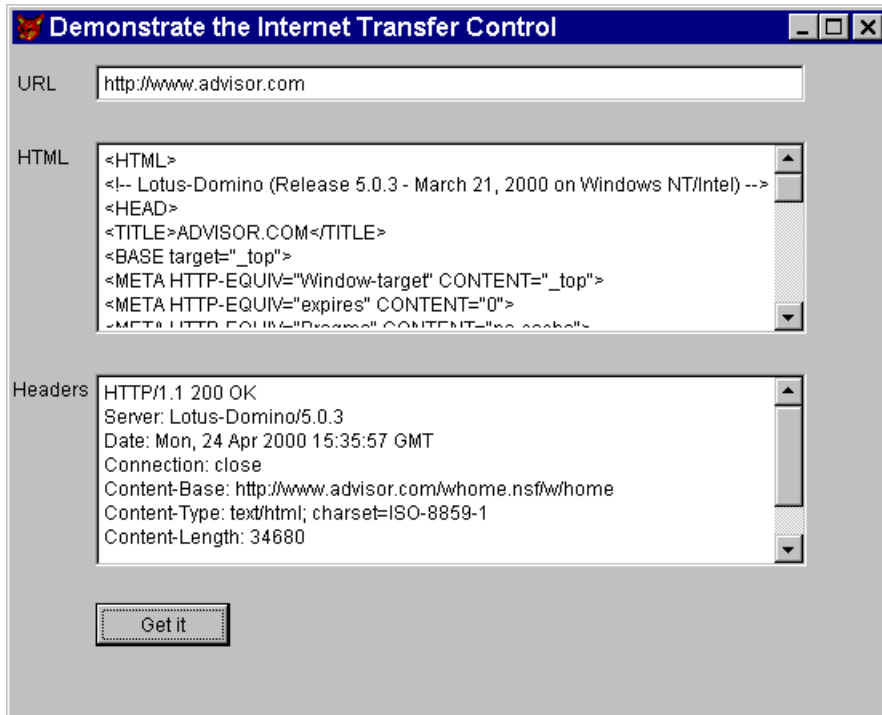


Figure 3 Using the Internet Transfer Control—Type a URL into the textbox and click the button and the contents of the specified page are transferred and displayed in the first editbox.

The form contains almost no code. It's all in the Click method of the button:

```
ThisForm.cHTML = ;
  ThisForm.OleInetControl.Object.OpenURL(;
    ALLTRIM(ThisForm.TxtURL.Value))
ThisForm.edtHeaders.Value = ;
  ThisForm.oleInetControl.Object.GetHeader()
ThisForm.Refresh()
```

cHTML is a custom property of the form, added to hold the result of the transfer. It's also the ControlSource for the first editbox. When the user clicks, the OpenURL method of the Internet Transfer Control is called; the value of the textbox is trimmed and passed as a parameter. (Without trimming, the call doesn't produce an error, but doesn't produce any results either, because the URL can't be identified.)

The next line demonstrates the control's GetHeader method. This method lets you retrieve additional information from the server. If you call it without a parameter, it returns all the header information the server is willing to provide. If you pass it the name of the particular header you're interested in, it returns that information. Beware:

different servers provide different information. Asking for a header that a server doesn't offer raises an error.

I was unable to reproduce your crash. However, when I attempted to test both your program and my form in VFP 5.0a, the Execute line of your code and the OpenURL call in mine failed with an OLE error "Invalid Callee." Calling SYS(2333,0) in the Load method of the form (as per Knowledge Base article Q170752) solved that problem and the form worked in that version as well. (SYS(2333) was added in VFP 5 just for situations like this one.)

What I've shown here barely scratches the surface of what you can do with the Internet Transfer control. In particular, I haven't discussed at all the control's FTP abilities. The Help file that comes with it provides a good start. To access it, right-click on the control in the Form Designer and choose Help. In addition, there are a number of relevant articles on the MSDN website. Search for "Internet Transfer Control".

-Tamar