

ActiveX control in python vs ActiveX control in vb 6 (piece of code)

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2007-08/msg02589.html>

- *From:* vml <victor.lebrun@xxxxxxxxxx>
 - *Date:* Wed, 22 Aug 2007 09:57:25 -0000
-

Hello,

I am trying to mograte from vb 6 to python.

I have a very usefull AX control. It can be :

- just a bitmap
- an active picture where you can plot data and put cursors
- a data explorer

Those 3 behavior are driven by one property in visual studio when you insert the AX control in a form: the property ModuleName:

- 'default' for bitmap
- 'Picture' for a picture

I try to insert this AX control in a wxpython based application (I used one of the example delivered with wx) but when I try to change the property to get an active picture it does not work and display me the default bitmap :(

Here is the pice of code

```
#####
```

```
import wx

if wx.Platform == '__WXMSW__':
    from wx.lib.activexwrapper import MakeActiveXClass
    import win32com.client
    try:
        ocx = win32com.client.gencache.EnsureModule('{33924602-
        C29E-4915-91B1-767CA3D675C2}',0x0,1,0)

    except:
        raise ImportError("Can't load ocx")
```

ActiveX control in python vs ActiveX control in vb 6 (piece of code)

```
#-----  
  
class TestPanel(wx.Panel):  
    def __init__(self, parent, log):  
        wx.Panel.__init__(self, parent, -1)  
        self.pdf = None  
  
        sizer = wx.BoxSizer(wx.VERTICAL)  
        btnSizer = wx.BoxSizer(wx.HORIZONTAL)  
  
        # this function creates a new class that can be used as  
        # a wxWindow, but contains the given ActiveX control.  
  
        coclass=ocx.ActiveXModuleLoader  
  
        print coclass  
  
        coclass.ModuleName='picture'  
  
        ActiveXWrapper = MakeActiveXClass(coclass)  
  
        # create an instance of the new class  
        self.pdf = ActiveXWrapper(self)  
        #self.pdf.ModuleName='PictureManager'  
  
        sizer.Add(self.pdf, 1, wx.EXPAND)  
  
        btn = wx.Button(self, wx.NewId(), "Open PDF File")  
        wx.EVT_BUTTON(self, btn.GetId(), self.OnOpenButton)  
        btnSizer.Add(btn, 1, wx.EXPAND|wx.ALL, 5)  
  
        btn = wx.Button(self, wx.NewId(), "<-- Previous Page")  
        wx.EVT_BUTTON(self, btn.GetId(), self.OnPrevPageButton)  
        btnSizer.Add(btn, 1, wx.EXPAND|wx.ALL, 5)  
  
        btn = wx.Button(self, wx.NewId(), "Next Page -->")  
        wx.EVT_BUTTON(self, btn.GetId(), self.OnNextPageButton)  
        btnSizer.Add(btn, 1, wx.EXPAND|wx.ALL, 5)  
  
        btnSizer.Add((50, -1), 2, wx.EXPAND)  
        sizer.Add(btnSizer, 0, wx.EXPAND)  
  
        self.SetSizer(sizer)
```

ActiveX control in python vs ActiveX control in vb 6 (piece of code)

```
self.SetAutoLayout(True)

wx.EVT_WINDOW_DESTROY(self, self.OnDestroy)

def OnDestroy(self, evt):
    if self.pdf:
        self.pdf.Cleanup()
        self.pdf = None

def OnOpenButton(self, event):
    print 'open'

def OnPrevPageButton(self, event):
    print 'button'

def OnNextPageButton(self, event):
    print self.pdf.ModuleName

#-----

def runTest(frame, nb, log):
    if wx.Platform == '__WXMSW__':
        win = TestPanel(nb, log)
        return win
    else:
        dlg = wx.MessageDialog(frame, 'This demo only works on MSW.',
            'Sorry', wx.OK | wx.ICON_INFORMATION)
        dlg.ShowModal()
        dlg.Destroy()

overview = __doc__

#-----

if __name__ == '__main__':
    import sys
    class TestFrame(wx.Frame):
        def __init__(self):
            wx.Frame.__init__(self, None, -1, "ActiveX test",
                size=(640, 480),
                style=wx.DEFAULT_FRAME_STYLE|
                wx.NO_FULL_REPAINT_ON_RESIZE)
            self.tp = TestPanel(self, sys.stdout)
```

ActiveX control in python vs ActiveX control in vb 6 (piece of code)

```
app = wx.PySimpleApp()
frame = TestFrame()
frame.Show(True)
app.MainLoop()
```

#####

I suspect that the activex control is bugged but how to know that since it is working perfectly under visual studio vb 6 but not with python ?:(

please help me !

.