

Re: Running a script using VB6 ScriptControl

Source:

<http://www.tech-archive.net/Archive/Scripting/microsoft.public.scripting.vbscript/2005-07/msg00703.html>

- *From:* mr_unreliable <kindlyReplyToNewsgroup@xxxxxxxxxxxxx>
 - *Date:* Tue, 19 Jul 2005 17:46:21 -0400
-

hi Ron,

If you want to use WScript -- WRITE IT YOURSELF!!!

In other words, write some code to implement the members of wscript you wish to use, and then make THAT wscript available to your script code.

The attachment is an example of some vb class code that implements some (obviously not all) of the methods and properties found in wscript.

Note that you were looking for "wscript.sleep" and that one is included.

Finally, here is a snippet of code demonstrating how to make your home-brew wscript object available to your script.

```
--- <snip> ---
Private Sub Form_Load()
Dim oWScript As New clsWScript

    With scScript ' initialize the script control...
        .AllowUI = True ' allow script to display user interface (ex: MB)
        .Language = "VBScript"
    End With
```

Re: Running a script using VB6 ScriptControl

```
' set up the object reference(s)...  
scScript.AddObject "WScript", oWScript, True
```

```
....  
--- </snip> ---
```

cheers, jw

You got questions? WE GOT ANSWERS!!! ..(but,
no guarantee the answers will be applicable to the questions)

Ron Lessnick wrote:

First I apologize for cross posting this also on the vb.controls newsgroup,
I just wasn't quite sure which group would be better suited to handle my
question...

I'm running a script (set up as a string) internal in my VB6 program and
feeding it into the MS ScriptControl. All works well except that I need the
script to
"sleep" for a few seconds. If I wrote the script as a standard .vbs script
file outside of VB I would add a line like this - WScript.Sleep(3000) -
With the WScript object being the intrinsic built in object of the scripting
engine. However, adding that line to my script string in the VB6 program
throws rt error 424 "Object Required: WScript".

So it seems the intrinsic object WScript is not recognized when you run a
script using the VB ScriptControl.

Anybody know how to get around this problem so that the WScript object can
be used, or at least how to make the script "Sleep"

TIA

Re: Running a script using VB6 ScriptControl

Ron

```
VERSION 1.0 CLASS
BEGIN
    MultiUse = -1 'True
END
Attribute VB_Name = "clsWScript"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
' WScript Class, an attempt to provide WScript Functionality
' to scripts running in the ms ScriptConstrol, jw 09Jun01

Option Explicit

' --- Declarations and Constants -----
Private Declare Sub apiSleep Lib "kernel32" Alias "Sleep" (ByVal dwMilliseconds As Long)
'
Private Declare Function SetCurrentDirectory Lib "kernel32" _
    Alias "SetCurrentDirectoryA" (ByVal lpPathName As String) As Long
Private Declare Function GetCurrentDirectory Lib "kernel32" _
    Alias "GetCurrentDirectoryA" (ByVal nBufferLength As Long, ByVal lpBuffer As String) As Long
'
Private Const MAX_PATH = 260
'
Private m_myParent As frmMain ' parent of this class as form object
'
Private nRtn As Long ' used to hold api return values...
' --- end of decalarations and constants -----
'
' -----
' the following two routines are part of the "callback" process, whereby the
' parent (control) and the child (this class) exchange calling cards.
' The "callback" is used as a way to access code and objects in the parent form.
' For more info on this, get yourself a copy of Dan Appleman's book, entitled:
' "Developing ActiveX Components w/vb 5", pg 243.
' -----

Public Property Let Parent(pSource As frmMain)

    Debug.Print "[myForm] .. frmSetParent called"
    ' debugging stuff (bugassert expects TRUE)...
    ArgAssert Not (pSource Is Nothing), "[LetParent], pSource is nothing!!!"

    Set m_myParent = pSource

End Property

Public Property Get Parent() As frmMain

    Debug.Print "[myForm] .. frmGetParent called"
```

Re: Running a script using VB6 ScriptControl

```
' debugging stuff (bugassert expects TRUE)...
ArgAssert Not (m_myParent Is Nothing), "[GetParent], m_myParent is nothing!!!"

Set Parent = m_myParent

End Property
' --- end of callback stuff -----

Public Property Get CreateObject(vProgID As Variant, Optional vPrefix As Variant = "") As Object
    Set CreateObject = vbCreateObject(CStr(vProgID))

    ' if there is a second argument provided, then connect it...
    ' If IsMissing(vPrefix) Then Exit Property
    ' Parent.wsConnect CStr(vPrefix), CreateObject
End Property

' --- Get / Set Current Working Directory -----

Public Property Let CurrentDirectory(vRHS As Variant)
    nRtn = SetCurrentDirectory(CStr(vRHS))
End Property

Public Property Get CurrentDirectory() As Variant
Dim sBuf As String
Dim iNull As Integer

    sBuf = String(MAX_PATH, Chr(0)) ' allocate string buffer
    nRtn = GetCurrentDirectory(MAX_PATH, sBuf)

    iNull = InStr(sBuf, Chr(0)) ' trim the returned string...
    If iNull > 0 Then sBuf = Left(sBuf, iNull)
    CurrentDirectory = sBuf
End Property

' --- "Easy" WScript Methods (Echo, Sleep, Quit) ---

Public Sub Echo(sMsg As Variant)
    ' ignore any multiple-string-parameters, for now...
    MsgBox CStr(sMsg)
End Sub

Public Sub Sleep(tSnooze As Variant)
    apiSleep CLng(tSnooze)
End Sub

Public Sub Quit()
    Parent.QuitScript
End Sub
```