

Re: ChangeFileOpenDirectory

Source:

<http://www.tech-archive.net/Archive/Word/microsoft.public.word.vba.general/2007-11/msg00546.html>

- *From:* "Ed" <nospam@xxxxxxxxxx>
 - *Date:* Mon, 26 Nov 2007 05:24:25 -0500
-

Okay, I just tested with 'ScreenRefresh' in lieu of the speed bump. It didn't work. I must be something else, but the speed bump definitely works. Weird.
--Ed

"Russ" <drsN0SPAMmikle@xxxxxxxxxxxxxxxxxxxxxxxx> wrote in message <news:C36FA3D9.1F7CC%drsN0SPAMmikle@xxxxxxxxxxxxxxxxxxxxxxxx>

Ed,
I'm unclear on what originally caused your problem. Was it the time needed to build up and store the path string into the newpath variable or the ChangeFileOpenDirectory syntax command line?
You said you put a delay before the command, so I'm guessing it was the method you used to build up the newpath path.

VBA should inherently wait when calling one of its own subroutines. But if you go outside of VBA by using a Windows shell or script routine, VBA will happily initiate the shell/script stuff, but won't wait for any return values.

Here is some information from a past message thread on how to make VBA wait while going outside of VBA:
=====Quote
There are other ways to move files using other scripting languages, too. In a dos batch file, you could use xxcopy (a variation of xcopy).
<<http://www.xxcopy.com/xxcopy17.htm>>

You could call a dos batch file from VBA.
Helmut Weber mentioned this:
<<http://vb.mvps.org/samples/project.asp?id=Shell32>>

Or this xShell code works in Word97, too:

Re: ChangeFileOpenDirectory

Put this in Declarations section at the top of your VBA code module so that all subroutines can take advantage of the 'wait for shell' code.

```
Private Declare Function CloseHandle Lib "kernel32" ( _  
    ByVal hObject As Long) As Long  
Private Declare Function GetExitCodeProcess Lib "kernel32" ( _  
    ByVal hProcess As Long, lpExitCode As Long) As Long  
Private Declare Function OpenProcess Lib "kernel32" ( _  
    ByVal dwDesiredAccess As Long, ByVal bInheritHandle As Long, _  
    ByVal dwProcessId As Long) As Long
```

'Add this code as one of the regular subroutines.

```
Public Function ShellX( _  
    ByVal PathName As String, _  
    Optional ByVal WindowStyle As Integer = vbMinimizedFocus, _  
    Optional ByVal Events As Boolean = True _  
    ) As Long
```

'Declarations:

```
Const STILL_ACTIVE = &H103&  
Const PROCESS_QUERY_INFORMATION = &H400&  
Dim ProcId As Long  
Dim ProcHnd As Long
```

'Get process-handle:

```
ProcId = Shell(PathName, WindowStyle)  
ProcHnd = OpenProcess(PROCESS_QUERY_INFORMATION, True, ProcId)
```

'wait for process end:

```
Do  
If Events Then DoEvents  
GetExitCodeProcess ProcHnd, ShellX  
Loop While ShellX = STILL_ACTIVE
```

'clean up:

```
CloseHandle ProcHnd  
End Function  
+++++
```

'And call it like this:

```
Dim x As Long  
Dim strDosBatchFullPath As String  
strDosBatchFullPath = ^C:\...myDosBatchFile.bat^  
System.Cursor = wdCursorWait  
x = ShellX(Chr(34) & strDosBatchFullPath & Chr(34))  
=====UnQuote
```

Re: ChangeFileOpenDirectory

Russ,

You have hit the nail on the head. There was not enough time for the recalcitrant computer to resolve itself from the call to the sub-routine (module). This happened in several other situations where sub-routines were called. I found myself having to put 1/10th second counters in a half dozen (so far) locations to insure proper processing. Any idea why this is allowed to happen in a program such as VBA??

Ed (in Virginia)

"Russ" <drsN0SPAMmikle@xxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:C36A1363.1F5D6%drsN0SPAMmikle@xxxxxxxxxxxxxxxxxxxxxxxx

Hey Ed,
Helmut's suggestion seems sufficient.
However, I'm curious.
If you put the line:
MsgBox "<<" & newpath & ">>"
Just before the ChangeFileOpenDirectory line in the recalcitrant computer, does that message popup show the path string you expect between the chevrons and/or is that enough time to allow the path to resolve itself before the FileOpen dialog appears?

Hi Ed,

this one is working for me today,
but who knows about tomorrow?

```
Sub OpenFolder()  
Dim f1 As String  
Dim f2 As String  
Dim f3 As String  
f1 = "c:\test\word1\  
f2 = "c:\test\word2\  
f3 = "c:\test\word3\  
ChangeFileOpenDirectory f3  
With Dialogs(wdDialogFileOpen)  
.Name = f3 & "*.doc" ' ! Path & name
```

Re: ChangeFileOpenDirectory

```
.Show  
End With  
End Sub
```

Greetings from Bavaria, Germany

Helmut Weber, MVP WordVBA

Vista Small Business, Office XP

I have installations of the identical VBA program on 20 different computers. I use the following command with some frequency.

```
ChangeFileOpenDirectory  
newpath  
With  
Dialogs(wdDialogFileOpen)  
.name = "*.doc"  
CloseNum = .Show  
End With
```

The command works perfectly on 19. On the 20th it refuses to work. The MyDocuments folder consistently opens when the section of code is encountered.

I have doubled the "ChangeFileOpenDirectory newpath" command (I found that suggestion on a board) but that doesn't work either.

When I step through (debugging) the vba code it works fine, so I know that the connections are intact.

Re: ChangeFileOpenDirectory

Any thoughts?

Ed (in Virginia)

--
Russ

drsmN0SPAMikleAThotmailDOTcom.INVALID

--
Russ

drsmN0SPAMikleAThotmailDOTcom.INVALID