

## Re: References for VBA

---

*Source:*

<http://www.tech-archive.net/Archive/Word/microsoft.public.word.vba.userforms/2005-10/msg00085.html>

---

- *From:* "Tony Jollans" <No Mail>
  - *Date:* Tue, 18 Oct 2005 08:41:29 +0100
- 

Hi Amy,

You seem to be suggesting that because you've found one anomaly in Access where the help file wasn't entirely accurate that there is some kind of Office-wide implication. Don't you think that maybe that's jumping to conclusions?

Help is not entirely accurate all over the place in all of Office and elsewhere and, in general, Office help (pre-2003 anyway) is better than average. One can argue that it shouldn't be the case but the reality of the world is that documentation always comes second to functionality. One hopes that errors are corrected but the chances of that are higher if the failings are pointed out and a failing in Access help (and, even more so, a failing in Access itself) should surely be pointed out in an Access forum.

For what it's worth, it seems to me that it is probably Access, rather than the help, which is at fault. Office is gradually moving towards being a fully integrated suite rather than a collection of disparate programs and this seems to me to be a point where Access has yet to be fully brought into line.

For the record, it's not actually the FileDialog which doesn't work in Access, it's the mso constants that aren't defined, so ...

```
Set dlg = Application.FileDialog(3)
```

.. would have worked. That kind of thing is always worth checking when you have an error and, in this case it gives a good indication of the reference that might be missing.

Finally, I've not been here long and haven't been on any Access newsgroups so can't comment on the comparison but I'm happy that you find the people here knowledgeable and helpful; I do too.

--

Enjoy,  
Tony

Re: References for VBA

"Amy Blankenship" <Amy\_nospam@xxxxxxxxxxxxxxxxxxxxxxxx> wrote in message  
[news:Ot6jXa20FHA.3856@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:Ot6jXa20FHA.3856@xxxxxxxxxxxxxxxxxxxxxxxx)

> Yes, but you'd think the answer would have implications Office-wide.

There

> could just as easily be some code I insert into a Word UserForm that worked

> in Access that fails to work in Word because of a missing reference, or code

> that appears in the Help that fails to work out of the box in ANY office

> application.

>

> I asked here because for questions of this type the usual people who respond

> here give what I think are better answers than people answering the same

> type of question in Access forums, perhaps because most Access users have to

> do some form coding eventually, whereas only the top minority of Word users

> wind up doing user forms.

>

> -Amy

>

> "Tony Jollans" <No Mail> wrote in message

> [news:ucFuvO10FHA.1252@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:ucFuvO10FHA.1252@xxxxxxxxxxxxxxxxxxxxxxxx)

>> I think you should ask this in an Access newsgroup. The code works as

>> advertised in Word but does not work in Access because the Office library

>> is

>> not set as a reference by default – I do not know whether the help or the

>> application is at fault.

>>

>> --

>> Enjoy,

>> Tony

>>

>>

>> "Amy Blankenship" <Amy\_nospam@xxxxxxxxxxxxxxxxxxxxxxxx> wrote in message

>> [news:OqiDel00FHA.3756@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:OqiDel00FHA.3756@xxxxxxxxxxxxxxxxxxxxxxxx)

>>> My code:

>>> In Word:

>>>

>>> Private Sub OutputBrowse\_Click()

>>> Dim dlg As FileDialog

>>> Set dlg = Application.FileDialog(msoFileDialogFolderPicker)

>>> If GetDirectory.InputDirectory.Value <> "" Then

>>> dlg.InitialFileName = GetDirectory.InputDirectory.Value

>>> Else

>>> dlg.InitialFileName = "C:\"

>>> End If

## Re: References for VBA

```
>>> If dlg.Show = -1 Then
>>> OutputDirectory.Value = dlg.SelectedItems(1)
>>> Call EnableButton
>>> End If
>>> End Sub
>>>
>>> In Access:
>>>
>>> Sub GetFile(Control)
>>> Dim retFile As String, dlg As Variant, s As String
>>> Set dlg = Application.FileDialog(msoFileDialogFilePicker)
>>> With dlg
>>> .InitialFileName = CodeProject.Path
>>> If .Show = -1 Then s = .SelectedItems(1)
>>> End With
>>> If s <> "" Then
>>> retFile = Right(s, Len(s) - InStrRev(s, "\"))
>>> Control.Value = retFile
>>> End If
>>>
>>> The Help
>>> In Word:
>>> FileDialog Object
>>> Provides file dialog box functionality similar to the functionality of
>>> the
>>> standard Open and Save dialog boxes found in Microsoft Office
>>> applications.
>>> With these dialog boxes, users of your solutions can easily specify the
>>> files and folders that your solution should use.
>>>
>>> Using the FileDialog object
>>> Use the FileDialog property to return a FileDialog object. The
FileDialog
>>> property is located in each individual Office application's Application
>>> object. The property takes a single argument, DialogType, that
determines
>>> the type of FileDialog object that the property returns. There are four
>>> types of FileDialog object:
>>>
>>> a.. Open dialog box – lets users select one or more files that you
can
>>> then open in the host application using the Execute method.
>>> b.. SaveAs dialog box – lets users select a single file that you can
>>> then
>>> save the current file as using the Execute method.
>>> c.. File Picker dialog box – lets users select one or more files. The
>>> file
>>> paths that the user selects are captured in the FileDialogSelectedItems
>>> collection.
>>> d.. Folder Picker dialog box – lets users select a path. The path
that
```

## Re: References for VBA

```
>> the
>>> user selects is captured in the FileDialogSelectedItems collection.
>>> Each host application can only instantiate a single instance of the
>>> FileDialog object. Therefore, many of the properties of the FileDialog
>>> object persist even when you create multiple FileDialog objects.
>> Therefore,
>>> make sure that you've set all of the properties appropriately for your
>>> purpose before you display the dialog box.
>>>
>>> In order to display a file dialog box using the FileDialog object, you
>> must
>>> use the Show method. Once a dialog box is displayed, no code will
execute
>>> until the user dismisses the dialog box. The following example creates
>>> and
>>> displays a File Picker dialog box and then displays each selected file
in
>> a
>>> message box.
>>>
>>> Sub Main()
>>>
>>> 'Declare a variable as a FileDialog object.
>>> Dim fd As FileDialog
>>>
>>> 'Create a FileDialog object as a File Picker dialog box.
>>> Set fd = Application.FileDialog(msoFileDialogFilePicker)
>>>
>>> 'Declare a variable to contain the path
>>> 'of each selected item. Even though the path is a String,
>>> 'the variable must be a Variant because For Each...Next
>>> 'routines only work with Variants and Objects.
>>> Dim vrtSelectedItem As Variant
>>>
>>> 'Use a With...End With block to reference the FileDialog object.
>>> With fd
>>>
>>> 'Use the Show method to display the File Picker dialog box and
>>> return the user's action.
>>> 'The user pressed the action button.
>>> If .Show = -1 Then
>>>
>>> 'Step through each string in the FileDialogSelectedItems
>>> collection.
>>> For Each vrtSelectedItem In .SelectedItems
>>>
>>> 'vrtSelectedItem is a String that contains the path of
>>> each
>>> selected item.
>>> 'You can use any file I/O functions that you want to
work
```

## Re: References for VBA

```
>>> with this path.
>>> 'This example simply displays the path in a message
box.
>>> MsgBox "The path is: " & vrtSelectedItem
>>>
>>> Next vrtSelectedItem
>>> 'The user pressed Cancel.
>>> Else
>>> End If
>>> End With
>>>
>>> 'Set the object variable to Nothing.
>>> Set fd = Nothing
>>>
>>> End SubIn Access:
>>>
>>> FileDialog Object
>>> Provides file dialog box functionality similar to the functionality of
>>> the
>>> standard Open and Save dialog boxes found in Microsoft Office
>> applications.
>>> With these dialog boxes, users of your solutions can easily specify the
>>> files and folders that your solution should use.
>>>
>>> Using the FileDialog object
>>> Use the FileDialog property to return a FileDialog object. The
FileDialog
>>> property is located in each individual Office application's Application
>>> object. The property takes a single argument, DialogType, that
determines
>>> the type of FileDialog object that the property returns. There are four
>>> types of FileDialog object:
>>>
>>> a.. Open dialog box – lets users select one or more files that you
can
>>> then open in the host application using the Execute method.
>>> b.. SaveAs dialog box – lets users select a single file that you can
>> then
>>> save the current file as using the Execute method.
>>> c.. File Picker dialog box – lets users select one or more files. The
>> file
>>> paths that the user selects are captured in the FileDialogSelectedItem
>>> collection.
>>> d.. Folder Picker dialog box – lets users select a path. The path
that
>> the
>>> user selects is captured in the FileDialogSelectedItem collection.
>>> Each host application can only instantiate a single instance of the
>>> FileDialog object. Therefore, many of the properties of the FileDialog
>>> object persist even when you create multiple FileDialog objects.
>> Therefore,
```

## Re: References for VBA

```
>>> make sure that you've set all of the properties appropriately for your
>>> purpose before you display the dialog box.
>>>
>>> In order to display a file dialog box using the FileDialog object, you
>> must
>>> use the Show method. Once a dialog box is displayed, no code will
execute
>>> until the user dismisses the dialog box. The following example creates
>>> and
>>> displays a File Picker dialog box and then displays each selected file
in
>> a
>>> message box.
>>>
>>> Sub Main()
>>>
>>> 'Declare a variable as a FileDialog object.
>>> Dim fd As FileDialog
>>>
>>> 'Create a FileDialog object as a File Picker dialog box.
>>> Set fd = Application.FileDialog(msoFileDialogFilePicker)
>>>
>>> 'Declare a variable to contain the path
>>> 'of each selected item. Even though the path is a String,
>>> 'the variable must be a Variant because For Each...Next
>>> 'routines only work with Variants and Objects.
>>> Dim vrtSelectedItem As Variant
>>>
>>> 'Use a With...End With block to reference the FileDialog object.
>>> With fd
>>>
>>> 'Use the Show method to display the File Picker dialog box and
>>> return the user's action.
>>> 'The user pressed the action button.
>>> If .Show = -1 Then
>>>
>>> 'Step through each string in the FileDialog.SelectedItems
>>> collection.
>>> For Each vrtSelectedItem In .SelectedItems
>>>
>>> 'vrtSelectedItem is a String that contains the path of
>> each
>>> selected item.
>>> 'You can use any file I/O functions that you want to
work
>>> with this path.
>>> 'This example simply displays the path in a message
box.
>>> MsgBox "The path is: " & vrtSelectedItem
>>>
>>> Next vrtSelectedItem
```

Re: References for VBA

>>> 'The user pressed Cancel.  
>>> Else  
>>> End If  
>>> End With  
>>>  
>>> 'Set the object variable to Nothing.  
>>> Set fd = Nothing  
>>>  
>>> End SubNote that there is nothing in either help mentioning a needed  
>>> reference, yet in Word it "just works." In Access, it doesn't.  
>>>  
>>> -Amy  
>>>  
>>> "Jonathan West" <jwest@xxxxxxxx> wrote in message  
>>> [news:uw2oPRw0FHA.2064@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:uw2oPRw0FHA.2064@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)  
>>>>  
>>>> "Amy Blankenship" <Amy\_nospam@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in  
message  
>>>> [news:u2x8e\\$Z0FHA.3780@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:u2x8e$Z0FHA.3780@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)  
>>>>> But the code I used was in the Help for both Access and Word. In  
Word  
>>> it  
>>>>> just worked, whereas in Access I had to check various references  
until  
>>> I  
>>>>> could make it work. In neither place did it mention that any  
>>>> references  
>>>>> were needed. That's the point...I got the code out of the Help  
files  
>>> for  
>>>>> Word and I saw the identical code in the Help files in Access, but  
in  
>>>>> neither place did it mention that the functionality wouldn't just  
work  
>>>>> "out of the box." If it isn't Microsoft's responsibility to document  
>>>> what  
>>>>> you need to do to make the code in their own help files work, then  
>>>> whose  
>>>>> is it? Are developers supposed to just GUESS, like I did? That  
seems  
>>> to  
>>>>> be a bit of an odd attitude for a company the size of Microsoft to  
>>>>> take  
>>>>> for their products.  
>>>>  
>>>>  
>>>>> Show us the relevant code  
>>>>  
>>>>  
>>>>> --  
>>>>> Regards

>>>> Jonathan West – Word MVP  
>>>> www.intelligentdocuments.co.uk  
>>>> Please reply to the newsgroup  
>>>> Keep your VBA code safe, sign the ClassicVB petition  
www.classicvb.org  
>>>  
>>>  
>>  
>>  
>  
>

- 
- *Follow-Ups:*
    - ◆ **Re: References for VBA**  
◇ From: Amy Blankenship
  
  - *References:*
    - ◆ **References for VBA**  
◇ From: Amy Blankenship
    - ◆ **Re: References for VBA**  
◇ From: Jonathan West
    - ◆ **Re: References for VBA**  
◇ From: Jonathan West
    - ◆ **Re: References for VBA**  
◇ From: Amy Blankenship
    - ◆ **Re: References for VBA**  
◇ From: Jonathan West
    - ◆ **Re: References for VBA**  
◇ From: Amy Blankenship
    - ◆ **Re: References for VBA**  
◇ From: Jonathan West
    - ◆ **Re: References for VBA**  
◇ From: Amy Blankenship
    - ◆ **Re: References for VBA**  
◇ From: Tony Jollans
    - ◆ **Re: References for VBA**  
◇ From: Amy Blankenship
  
  - Prev by Date: **Bookmark Tool**
  - Next by Date: **Re: References for VBA**
  - Previous by thread: **Re: References for VBA**
  - Next by thread: **Re: References for VBA**
  - Index(es):
    - ◆ **Date**
    - ◆ **Thread**