

## Re: Use of CallByName()

---

*Source:*

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2008-08/msg04035.html>

---

- *From:* Dave Peterson <petersod@xxxxxxxxxxxxxxxxxxx>
  - *Date:* Sat, 23 Aug 2008 16:07:50 -0500
- 

I think it's a good point, but if I knew that there were two subroutines with the same name in different modules, I think I'd try to fix it.

Sooner or later, I'm gonna be lots happier if I fix the problem at the source. I hate having to make things more complex in other (usually lots of other) areas rather than just fixing the problem where it should be fixed.

"Rick Rothstein (MVP - VB)" wrote:

I don't disagree with you at all, that is for sure, but I do know there are programmers out there who will use the same subroutine name in different modules... some times inadvertently. For example, a programmer might create a module with routines for dealing with personal information and create a GetAddress subroutine there; then, 3 months later he/she might develop a module dealing with cell ranges and include a GetAddress subroutine there too. If they later decide to use both modules in a future workbook, Boom!, there it, the same name in the two different modules, created at different times, in the same workbook. I just thought it might be a good idea to warn those out there of the possible pitfall they could face.

Rick

"Dave Peterson" <petersod@xxxxxxxxxxxxxxxxxxx> wrote in message [news:48B058B2.C8891C6F@xxxxxxxxxxxxxxxxxxx](mailto:news:48B058B2.C8891C6F@xxxxxxxxxxxxxxxxxxx)

As a general rule, I would say that it's never good to have two subroutines in different modules with the same name.

Heck, it can be difficult to have two workbook projects with the same procedure name. If you look at some of the examples on Ron de Bruin's site, you'll see that he's change some names to be "more" unique.

"Sub RDB\_Merge()" instead of "Sub Merge()". Controls on toolbars can be confused with macros with the same name, too. (Well, the controls aren't

Re: Use of CallByName()

confused. But they may not do what you want.)

"Rick Rothstein (MVP – VB)" wrote:

Interesting that this line works (I learned something new – thanks)...

```
Application.Run "" & ThisWorkbook.Name & ""!"" &  
SubNames(i)
```

This also works (it is the method I am familiar with)...

```
Application.Run "Module1." & SubNames(i)
```

assuming the OP was trying to call subroutines on a Module named Module1.

Your call line is more universal than mine in that it will call a subroutine from any module in the workbook as long as the subroutine names on each module are unique. If you had, say, two subroutines named Sub1, one on Module1 and the other on Module2, your code line would error out unless the programmer specifically specified which Module's Sub1 he/she wanted. That is, the assignment of the subroutine names for this situation would have had to be (assuming the Sub1 on Module1 was wanted)...

```
SubNames = Array("Module1.Sub1", "Sub2", "Sub3")
```

Sub2 and Sub3 could be on any module (Module1, Module2, etc.) because (I'm assuming here that) their subroutine names are not used on any other module than the one they are declared in.

Rick

"Dave Peterson" <petersod@xxxxxxxxxxxxxxxxxxxx> wrote in message  
[news:48B0425D.FA5A8D6D@xxxxxxxxxxxxxxxxxxxx](mailto:news:48B0425D.FA5A8D6D@xxxxxxxxxxxxxxxxxxxx)

## Re: Use of CallByName()

I'd drop the callbyname attempt and use application.run

```
Option Explicit
Sub testme()
Dim SubNames As Variant
Dim i As Long
SubNames = Array("Sub1", "Sub2", "Sub3")
For i = LBound(SubNames) To
UBound(SubNames)
Application.Run "" & ThisWorkbook.Name
& "!" & SubNames(i)
Next i
End Sub
Sub sub1()
MsgBox "sub1"
End Sub
Sub sub2()
MsgBox "sub2"
End Sub
Sub sub3()
MsgBox "sub3"
End Sub
```

George B wrote:

Thanks for your reply, but it doesn't seem to help me. In your example, the object is a textbox. What I need to know is how to do the following:

```
Replace:
Call Sub1
Call Sub2
Call Sub3
```

```
By:
subnames=array("Sub1","Sub2","Sub3")
For i=0 to 2
Call
someobject,subnames(i),vbMethod
Next i
```

I don't know how to specify someobject.

Re: Use of CallByName()

"Joel"

<Joel@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote in message

[news:A408A937-35A3-47A1-93DD-D855FDDEB42F@xxxxxxxxxxxxxxxx](mailto:news:A408A937-35A3-47A1-93DD-D855FDDEB42F@xxxxxxxxxxxxxxxx)

The code  
below  
works in  
2003 in a  
module. I  
took the  
help  
example  
in  
VBA

and

made a  
minor  
change. I  
also added a  
textbox to  
the active  
worksheet.  
See

if

this helps.

```
Set Text1 =  
ActiveSheet.OLEObjects("Textbox1").Object  
CallByName  
Text1,  
"MousePointer",  
VbLet,  
vbCrosshair  
Result =  
CallByName(Text1,  
"MousePointer",  
VbGet)
```

"George B"  
wrote:

All  
the

Re: Use of CallByName()

references  
I  
have  
found  
on  
this  
procedure  
appear  
to  
be  
for  
use

in a

class  
module.  
I  
would  
like  
to  
use  
it  
to  
call  
a  
procedure  
in  
Module1.

Is

this  
possible?  
What  
is  
the  
object  
to  
be  
specified  
in  
CallByName  
object,procname,calltype,[args()]

--

Re: Use of CallByName()

Dave Peterson

--

Dave Peterson

--

Dave Peterson

.