

Re: Auto Load Add-In

Source:

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2007-08/msg03697.html>

- *From:* "Peter T" <peter_t@discussions>
 - *Date:* Thu, 23 Aug 2007 09:48:07 +0100
-

Your code looks fine to add a project reference, in so doing will 'by default' load the file if not already loaded. I don't see anything in what you posted that will unload the addin.

The "addins" collection is in effect all the addins in the "Addins Manager" as listed in Tools > Addins, irrespective as to whether the addin is loaded (installed). The collection may also include some addins that do are not visible in the list if they are not installed and not in one of the default addin's paths.

```
Sub ListAddins()  
Dim i As Long  
Dim adn As AddIn  
  
Set adns = Application.AddIns  
  
For Each adn In Application.AddIns  
i = i + 1  
With adn  
Cells(i, 1) = .Installed ' ie ticked in Tools > Addins & loaded  
Cells(i, 2) = .Title ' File > Properties > Title  
Cells(i, 3) = .Name  
  
s = .Path & "\"  
If Len(s) <= 1 Then s = "MS built-in addin"  
Cells(i, 4) = s  
End With  
Next  
Columns("B:C").EntireColumn.AutoFit  
End Sub
```

As I've tried to explain, if you want to do .Installed = True, first you need to verify the addin exists in the addins collection and if not '.Add' it to the collection.

For your purposes I assume it's not necessary for your addin to exist in the addins collection as you are loading/unloading it by other means.

Re: Auto Load Add-In

I also assume the reason you want to add a project reference to the addin to your main wb, is because vba-code in your main wb directly calls code in the addin virtually as if the addin's code exists in the main-wb (that's the only reason to add the ref as you are doing).

Regards,
Peter T

"scott" <sbailey@xxxxxxxxxxxxxxxx> wrote in message
news:ujplJLS5HHA.2312@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Finally got it. I had to re-write it a little, but this will load/unload
and

test properly.

One final question:

Please explain what you meant by "addin actually in the addins collection, your code does not check for that. It needs to be before you can set the Installed property = true"

I don't exactly understand what the "addins" collection really means. And BTW, thank you for your input on this issue.

CODE: *****

```
Sub loadAddIn()  
  
Dim bFound As Boolean  
  
On Error Resume Next  
  
For Each Ref In ThisWorkbook.VBProject.References  
i = i + 1  
If Ref.Name = "test_AddIn" Then  
MsgBox "Test Add-In installed."  
bFound = True  
Exit For  
End If  
  
Next Ref  
  
If bFound = False Then _  
ThisWorkbook.VBProject.References.AddFromFile _  
"C:\data\test_addin.xla"  
  
On Error GoTo 0  
  
End Sub
```

Re: Auto Load Add-In

"Peter T" <peter_t@discussions> wrote in message
news:%23XzhjwR5HHA.1164@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Is the addin actually in the addins collection, your code does not check for that. It needs to be before you can set the Installed property = true. Refer to the notes I posted previously.

Why do you think excel is preventing me to simply test if a add-in

reference

exists?

What kind of reference are you referring to.

Where is your error generated.

Regards,
Peter T

"scott" <sbailey@xxxxxxxxxxxxxxxxxx> wrote in message
[news:e9\\$fSoR5HHA.1188@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:e9$fSoR5HHA.1188@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

I did manage to load the add-in into referencds after checking the

option

under Tools|Macro that gives access to vba projects. However, no matter if the add-in is loaded or not, I can't test it with vba. The code in

CODE

2 below continues to give the subscript out of range error.

Why do you think excel is preventing me to simply test if a add-in

reference

Re: Auto Load Add-In

exists?

CODE 1: *****

```
Sub loadAddIn()
```

```
ThisWorkbook.VBProject.References.AddFromFile _  
"c:\data\test_addin.xla"
```

```
End Sub
```

CODE 2: *****

```
Sub TestAddIn()
```

```
Dim tempStr As String  
tempStr = "C:\data\test_addin.xla"  
If Dir(tempStr) = "" Then  
MsgBox "you do not have test_addin.xla installed"  
End  
End If
```

```
If AddIns("test_AddIn").Installed = False _  
Then AddIns("test_AddIn").Installed = True
```

```
Dim bFound As Boolean  
Dim obj  
'see if there is a reference already  
For Each obj In ThisWorkbook.VBProject.References  
MsgBox "name: " & UCase(obj.Name)  
If UCase(obj.Name) = "test_addin.xla" Then  
bFound = True  
Exit For  
End If  
Next obj
```

```
'if no reference then set a reference.  
If bFound = False Then _  
ThisWorkbook.VBProject.References.AddFromFile _  
"C:\data\test_addin.xla"  
End Sub
```

ERROR 2: *****

Run-time error '9':
Subscript out of range

Re: Auto Load Add-In

"Peter T" <peter_t@discussions> wrote in message
news:uX0vJcR5HHA.3940@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

You're loosing me! Is this thread about loading an addin, or adding a references.

Anyway –

```
ThisWorkbook.VBProject.References.AddFromFile  
–  
"C:\data\test_addin.xla"
```

The above looks fine, assuming the ref does not already exist and a

valid

project in the path.

```
Dim tempStr As String  
tempStr =  
"C:\data\test_addin.xla"  
If Dir(tempStr) = "" Then  
MsgBox "you do not have  
test_addin.xla installed"  
End  
End If
```

The message implies the addin is not in the addins collection rather

than

not found in the specified folder.

Avoid the "End" statement. If necessary release any object variables

and

use Exit Sub or Function, possibly all the way back up to the

original

Re: Auto Load Add-In

calling routine.

```
If  
AddIns("test_AddIn").Installed  
=
```

Before the above, unless certain the addin
exists in the addins

collection

do -

```
On error resume next  
Set adn = addins(the-Addin's-Title)  
If adn is nothing then  
Set adn =  
addins.add(C:\data\test_addin.xla")  
End if
```

If not adn.installed then adn.installed = True

Normally once the reference has been added
to a wb, when the wb opens

the

referenced wb will open automatically.

Regards,
Peter T

"scott" <sbailey@xxxxxxxxxxxxxxxx> wrote
in message
news:O3iuPgQ5HHA.1208@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

I re-wrote my code into
several subs listed below. I'd
really like

to

learn

the method of adding the
addin to the references. Do

Re: Auto Load Add-In

Re: Auto Load Add-In

you have any

code

to

accomplish this?

I tried the sub loadAddIn()
below IN CODE #1 and I
get ERROR #1. If

I

call
the sub, TestAddIn() in
CODE #2, I get ERROR #2.
Excel won't even
allow
my
code in CODE #2 to test if
the add-in is loaded.

Basically, I'm striking out
no matter what I try. I know
there must

be

a

way

to accomplish this feat.

Note: I did manage to
properly name my project
"test_AddIn" instead

of

the
default "VBAProject".

CODE 1:

Sub loadAddIn()

ThisWorkbook.VBProject.References.AddFromFile

Re: Auto Load Add-In

```
–  
"C:\data\test_addin.xla"  
End Sub
```

ERROR 1:

Run-time error '1004':
Method 'VBProject' of
object '_Workbook' failed

CODE 2:

```
Sub TestAddIn()
```

```
Dim tempStr As String  
tempStr =  
"C:\data\test_addin.xla"  
If Dir(tempStr) = "" Then  
MsgBox "you do not have  
test_addin.xla installed"  
End  
End If
```

```
If  
AddIns("test_AddIn").Installed  
= False _  
Then  
AddIns("test_AddIn").Installed  
= True
```

```
Dim bFound As Boolean  
Dim obj  
'see if there is a reference  
already  
For Each obj In  
ThisWorkbook.VBProject.References  
MsgBox "name: " &  
UCase(obj.Name)  
If UCCase(obj.Name) =  
"test_addin.xla" Then  
bFound = True  
Exit For  
End If  
Next obj
```

```
'if no reference then set a  
reference.
```

Re: Auto Load Add-In

```
If bFound = False Then _  
ThisWorkbook.VBProject.References.AddFromFile  
_  
"C:\data\test_addin.xla"  
End Sub
```

ERROR 2:

Run-time error '9':
Subscript out of range

"Peter T"
<peter_t@discussions>
wrote in message
news:%23KUmcmQ5HHA.4164@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Looks like
you don't
particularly
need to add
the addin to
the

addin's

collection
so just open
and close it
like a
normal
workbook,
eg

```
Sub  
OpenCloseAddin(bOpen  
as boolean)  
dim sFile as  
string  
dim wb as  
workbook
```

```
sFile =  
"C:\data\test_addin.xla"  
On error  
resume next  
set wb =
```

Re: Auto Load Add-In

```
application.workboooks(sFile)
On error
goto errH
if bOpen
then
if wb is
nothing
then
application.workbooks.open
sFile
end if
elseif not
wb is
nothing
then
wb.close
end if
exit sub
errH:
if bOpen
then
msgbox
"Problem
opening " &
sFile
end if
end sub
```

warning –
aircode !

Regards,
Peter T

"scott"
<sbaily@xxxxxxxxxxxxxxxx>
wrote in
message
news:#CwdemP5HHA.4712@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

I'm
trying
to
load
an
add-in
when
my
workbook
opens.
First

Re: Auto Load Add-In

I
created
an
add-in
at
"C:\data\test_addin.xla"
that
contains
the
code
found

below

in

"CODE
2"
section.
Next,
I
added
code
found
below
in
"CODE
1"
section
to

the

file
"C:\data\test.xls".
While
the
add-in
was
open,
in
the

Module

property,

I
named
the
module

Re: Auto Load Add-In

Re: Auto Load Add-In

"test_AddIn".

If
I
open
"C:\data\test.xls"
containing
"CODE
1",
I
get
a

"Subscript

out

of

range"
error
and
trace
the
error
to
the
line

If
AddIns("test_AddIn").Installed
=
False

in
"CODE
1".
To
further
troubleshoot,
I
manually
added
the
add-in

within

Excel's
References
and

Re: Auto Load Add-In

noticed
the
name
of
my
add-in
within

References

is

called
"VBAProject".

I
assume,
the
error
is
being
caused
because
of
the
"VBAProject"

name

of

my

add-in.

QUESTIONS:

1.
How
can
I
change
the
"VBAProject"
name
to
"test_AddIn".
2.
Will
changing
the

Re: Auto Load Add-In

name
as
described
in
QUESTION
#1
allow
my

add-in

to

load
when
the
"C:\data\test.xls"
workbook
opens?
3.
If
I'm
going
about
this
wrong,
can
someone
shed
some
light
on
how
to
accomplish
this
task?

'
CODE
1:

Sub
Auto_Open()

Dim
tempStr
As

Re: Auto Load Add-In

```
String
tempStr
=
"C:\data\test_addin.xla"
If
Dir(tempStr)
=
""
Then
MsgBox
"You
do
not
have
the
Test
Add-In
installed."
End
End
If

If
AddIns("test_AddIn").Installed
=
False
-
Then
AddIns("test_AddIn").Installed
=
True

End
Sub

Sub
Auto_Close()

If
AddIns("test_AddIn").Installed
=
True
-
Then
AddIns("test_AddIn").Installed
=
False

End
Sub
```

Re: Auto Load Add-In

```
'  
CODE  
2:  
*****  
  
Sub  
Auto_Open()  
MsgBox  
"Add-In  
Loaded"  
End  
Sub
```