

Re: Help end the testing tedium please

Source:

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2007-10/msg03840.html>

- *From:* "Dean" <whooshbopbang4@xxxxxxxxxxxxx>
 - *Date:* Tue, 23 Oct 2007 11:09:03 -0700
-

I will try some of this. I wanted to let you know that I changed the 60 to 4, because I wanted to see the result with only 4 before I bothered to create 60 files, which is tedious. And since your message box only popped up once, and had the correct info for the first file, I assume it failed on the first file.

You didn't answer my questions as to whether, if a macro bombs, if the stuff that before the bomb stays put, or if it somehow is reset back to the beginning. I don't see that anything in column I of Utility is being filled in. More importantly, I was hoping you might be able to tell me how that column I of Utility gets populated. It seems that, perhaps, it is getting populated via the Consolidate Project macro and we seem to have circumvented that macro, but I'm not even sure that's where it happens. To me, it seems that (my "Then B = A "argument) Prj_Sht is set equal to what's in column I, not the other way around, and I'm still wondering how column I gets populated. When the macro starts, it is empty – I can tell you that. Do you have any idea? I think that is the problem.

Dean

"Joel" <Joel@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message <news:E1E203CA-5377-4299-AA74-D914F9ABD1B4@xxxxxxxxxxxxxxxxxxxxx>

Dean: Last night I tried to take the approach that would have the best change of working. I guess I failed.. I suspect you are getting error 1004 which indicates the workshhet was not found. I wondering if the code is failing when n = 1 1 or at some higher value of N. The code your have posted wqs checking for projects that have empty strings. Maybe it is failing because of a project that doesn't exist?

1) First try this change

```
from  
Call Consolidate(ProjectFile,PrjName)  
to  
Thisworkbook.activate
```

Re: Help end the testing tedium please

Call Consolidate(ProjectFile,PrjName)

The codee may be working for the first project and not the second. When a workbook opens occurs it changes the focus window to the newly opened workbook. the coede may be failing because the workbook that contains the worksheett Utility is not activated.

The real fix would be to change the statement that is failing but againn I don't want to risk breaking working code

```
from  
Sheets(Prj_Sht).Activate  
to  
Thisworkbook.Sheets(Prj_Sht).Activate
```

2) Try adding amessage box before this line of code and run the old code and see what you get. compare the results with the old code and new code

```
from  
Sheets(Prj_Sht).Activate  
to  
msgbox(Prj_Sht)  
Sheets(Prj_Sht).Activate
```

3) Look at the utility worksheet in columns G and I like the code below

```
If Sheets("Utility").Range("I" & (n + 40)) > "" Then  
Prj_Sht = Sheets("Utility").Range("I" & (n + 40))  
Else  
Prj_Sht = Sheets("Utility").Range("G" & (n + 40))  
End If
```

Code that checks if the project exists. this is more complicated and dfon't really want you to try this until we get the other code working. We may have to use it to skip projects that don't exist.

```
Sub Consolidate_All()  
Dim n As Integer  
MyPath = "C:\Documents and Settings\Dean\My Documents\AW\CONSOL  
TEMPLATE\files\  
  
Files = "test"  
For n = 1 To 60  
ThisWorkbook.Activate  
If Sheets("Utility").Range("I" & (n + 40)) > "" Then  
Prj_Sht = Sheets("Utility").Range("I" & (n + 40))  
Else  
Prj_Sht = Sheets("Utility").Range("G" & (n + 40))  
End If
```

Re: Help end the testing tedium please

```
If Prj_Sht <> "" Then
ProjectFile = MyPath & Prj_Sht & Files & n & ".xls"
'remove msgbox after code works
MsgBox ("File and project name are as follows : " & Chr(13) & _
ProjectFile & Chr(13) & Prj_Sht)
Call Consolidate(ProjectFile, Prj_Sht)
End If
Next n
End Sub
```

Last night I first stated writing the code like the code below but felt It would be harder to get working. Maybe we need to change the code.

"Dean" wrote:

The project names were, indeed, range names. I'm not sure that is important. The new macro is crashing (subscript out of range) at:

```
Sheets(Prj_Sht).Activate
```

I'm not sure why. The filename and sheet name look fine. It seems that we are now skipping the Consolidate Project macro and, perhaps, it was the last part of that macro that was defining what Prj_Sht is. Which brings me to that question that confuses you on the B = A thing: I can't figure out how column I of Utility gets populated. I know that, somehow, when the old macro ran, it did, but I can't figure out how.

By the way, if a macro crashes at a certain point, does everything before that point get done, or does it, somehow, not do anything? The cell on Utility is not getting populated – I can tell you that.

Thanks!
Dean

"Joel" <Joel@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:1E723878-04D3-403D-8175-339805111697@xxxxxxxxxxxxxxxxxxxx

1) The old code was/is working fine, so the problem is in the new code.
I fixed the wrap around problem as you advised. I am a little confused
by
the

Re: Help end the testing tedium please

two arguments needed for "Consolidate". When you call it from Consolidate All, it seems that you may be reversing their order from what Consolidate wants. Is that correct?

Is did reverse order. sorry

2) Considering the above, I am also a bit confused by your suggestion from automating the assumed filename, perhaps because you have copied the routine to name the project sheet. Please assume that both the filename and the worksheet name are sequential. Let's say the filenames are test1 thru test60 and that the sheet names will be joe jones test1 thru joe jones test 60

New code above is using your suggestion

3) Lastly, though it is not essential info for me to understand, I know that a worksheet called Utility somehow captures the sheet names, but am confused as to how it gets them. There are statements such as

```
If Sheets("Utility").Range("I" & (n + 40)) > "" Then  
Prj_Sht = Sheets("Utility").Range("I" & (n + 40))  
Else
```

Your original code had the function below which may be causing part of your problems. There is a names range called "projectname". I think you can see it in the worksheet menu

Insert – Name – Define

Re: Help end the testing tedium please

if it is not in the above try

File – Properties

Let me know which one it is located. There are few ways of defining ranges and I can't tell from the code which method is being used.

```
Sub Load_Consolidate_Dialog()  
'If this bombs, you may have to clear out the names in the  
hidden  
"utility"  
worksheet  
Dim n As Integer  
For n = 1 To 60  
If Range("projectName" & (n)).Value > "" Then  
Consolidate_Dialog.Controls("ProjectButton" & (n)).Caption  
= Range("ProjectName" & (n)).Value  
Else  
Consolidate_Dialog.Controls("ProjectButton" & (n)).Caption  
= "Project " & (n)  
End If  
Next n  
Consolidate_Dialog.Show  
End Sub
```

```
4) If Sheets("Utility").Range("I" & (n + 40)) > "" Then  
Prj_Sht = Sheets("Utility").Range("I" & (n + 40))  
Else
```

This code is a little different from Load_Consolidate_Dialog. It doesn't use a named range instead it is actually using a cell location in worksheet utility. Range("I55"). Notice the double quotes around the range which indicates this is a string. An ampersand (&) connects string together and converts a number (n) to a string. So if N = 55 (N is a number), Range("I" & N) is the same as Range("I55"). VBA converted

Re: Help end the testing tedium please

automatically the number
55
to
a string "55".

5) in the macro which are probably doing it, but I think I am confused because I am still interpreting them backwards. When I see "Then B = A", I assume that it means to assign A's value to B. But I am guessing that this is just the opposite. If so, I guess you have answered my question. If not, then if you have any clue as to how it is finding its way onto the Utility worksheet, let me know (it may be off somewhere in another macro). I do know which cell in each input file becomes Prj_Sht. I just am a little confused as to where the macro puts it.

B = A does mean assign A's value to B. Not sure why you are interpreting this backwards. It may be the IF statement

```
If Sheets("Utility").Range("I" & (n + 40)) > "" then
```

The above is a test check if the string on sheet utility Range "I?" (? is number) is not a blank string. I prefer not to use > but use <> (not equal)

```
If Sheets("Utility").Range("I" & (n + 40)) <> "" then
```

The statement below is the assignment if the string is not empty

```
Prj_Sht = Sheets("Utility").Range("I" & (n + 40))
```

New Consolidate All

I added a message box to help get it working. You can remove

Re: Help end the testing tedium please

the message
box
after we get the code working. Check the placement of the
spaces in
the
msgbox. You had "test60" and "test 60". Not sure which is
right. I
add
a
blank before the number (" "). Remove the blank and one of
the
ampersands
if
it is wrong from both statements below. chr(13) put a
carriage return
into
the messbox to make the message easier to read.

```
Sub Consolidate_All()  
Dim n As Integer  
mypath = "C:\Documents and Settings\Dean\My  
Documents\AW\CONSOL  
TEMPLATE\files\  
sht = "joe jones"  
files = "test"  
For n = 1 To 60  
PrjName = sht & " " & n  
ProjectFile = mypath & sht & files & " " & n & ".xls"  
'remove msgbox after code works  
msgbox("File and project name are as follows : " & chr(13)  
& _  
ProjectFile & chr(13) & PrjName)  
Call Consolidate(ProjectFile,PrjName)  
Next n  
End Sub
```

"Dean" wrote:

The old code was/is working fine, so the
problem is in the new code.
I
fixed the wrap around problem as you
advised. I am a little confused
by
the
two arguments needed for "Consolidate".
When you call it from
Consolidate
All, it seems that you may be reversing their
order from what

Re: Help end the testing tedium please

Consolidate
wants. Is that correct?

Considering the above, I am also a bit
confused by your suggestion fro
automating the assumed filename, perhaps
because you have copied the
routine
to name the project sheet. Please assume that
both the filename and
the
worksheet name are sequential. Let's say the
filenames are test1 thru
test60
and that the sheet names will be joe jones
test1 thru joe jones test
60

Lastly, though it is not essential info for me
to understand, I know
that
a
worksheet called Utility somehow captures
the sheet names, but am
confused
as to how it gets them. There are statements
such as

```
If Sheets("Utility").Range("I" & (n + 40)) >  
"" Then  
Prj_Sht = Sheets("Utility").Range("I" & (n +  
40))  
Else
```

in the macro which are probably doing it, but
I think I am confused
because
I am still interpreting them backwards.
When I see "Then B = A", I
assume
that it means to assign A's value to B. But I
am guessing that this
is
just
the opposite. If so, I guess you have
answered my question. If not,
then
if you have any clue as to how it is finding
its way onto the Utility
worksheet, let me know (it may be off
somewhere in another macro). I

Re: Help end the testing tedium please

do
know which cell in each input file becomes
Prj_Sht. I just am a
little
confused as to where the macro puts it.

Can you kindly answer these questions and
rewrite Consolidate ALL so
that
it
automates both the filename and the
worksheet name?

Thanks much for your help!
Dean

"Joel"
<Joel@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in message
news:5BE7DC8E-88D1-4603-8D95-FA49B66AADE8@xxxxxxxxxxxxxxxxxxxx

Dean: I trying to minimize
the risk of breaking the code
that is
already
working. I don't think you
have posted all the code
which I don't
need
but
don't want to break. There is
probably code on the sheet
pages and
Thisworkbook that you are
not aware of. Lets not worry
about that
code.
Lets just get your test
function working. From
your psting I'm
not
sure
if
the old code still works or
the problem is the new code.
right now
I'm
assuming the new code had
problems.

Re: Help end the testing tedium please

I modified my code to
include the path

```
Sub Consolidate_All()  
mypath = "C:\Documents  
and Settings\Dean\My  
Documents\AW\CONSOL  
TEMPLATE\files\"
```

```
Dim n As Integer  
ProjectFile =  
Array("", "Filename1", "Filename2", .....)  
For n = 1 To 60  
ProjectName =  
Range("projectName" &  
(n)).Value  
Call  
Consolidate(ProjectName,  
mypath & ProjectFile(n))  
Next n  
End Sub
```

Answer to questions

1) in the array, do the
filenames need the .xls
extension?

Yes

2) if the names of
the file really were test1 thru
test60, would the array
statement

need

them

all listed or is it smart
enough for a ... to work?

The are lots of ways to get
the filename. The array
method will
work
ynder
any condition. If there is a
pattern to the file name then
the
array
may
not
be necessary such like the
code below

Re: Help end the testing tedium please

```
For n = 1 To 60
ProjectName =
Range("projectName" &
(n)).Value
Call
Consolidate(ProjectName,
mypath & "test" & n)
Next n
```

3) By path, I assume you mean the path to the folder where the files will be. If so, I think we can ignore that for now, i.e., as long as the last time it was queried, we supplied the same folder, it will use that folder as a default, right?

The new code I gave you now includes the path

4) When you say:

I will make the ProjectFile Array easier to enter the names (they will be shorter).

I don't know what you mean. Are you planning on doing some more work?

As I said before, There are lots of ways to get the files names depending if there is a pattern to the names or if they are random. I wanted yo to see

Re: Help end the testing tedium please

what I was doing and then
expected exactly the
questions you are
asking.

the additional work is what
I'm doing now which is
helping you find
the
best way to get the
filenames. I hope I'm not
confusing you. I
felt
it
was
better not to give you too
much information until you
saw the code.
the
I
planned to answer the
questions you needed
answered.

5) I tried copying your
macro into my file, just
using, test1,
test2,
and
test3
as the three project names in
the array and I notice that,
every
time
the

```
Selection.PasteSpecial  
Paste:=xlValues,  
Operation:=xlNone,  
SkipBlanks:=  
-  
False, Transpose:=False
```