

## RE: Message boxes

---

*Source:*

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2007-07/msg00196.html>

---

- *From:* Jock <[Jock@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Jock@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 2 Jul 2007 09:32:03 -0700
- 

Thanks Steve – great job.  
The code for this project is starting to look like a novel.  
Will tackle this tomorrow.

--  
tia

Jock

"Steve the large" wrote:

I meant where you set the flag false.

"Steve the large" wrote:

easy peasy...

under the line where you set the flag true add

```
TB.backcolor= RGB(255,255,0)
count= count +1
```

I think this is yellow, I think white is RGB(255,255,255).

Then for each empty box the color changes. Then the reminder message can be:

```
msgbox Prompt:="lease fill in the " & str$(count) & " highlighted boxes"
```

Notes:

You can only see the background color of an object if the BackStyle property is set to fmBackStyleOpaque.

add a "Dim count as long" and initialize count to zero before looping through the TB's

## RE: Message boxes

You need to clear all six bgcolor properties in the form open event.

You need to clear the bgcolor property whenever the text is entered in the text box change event (if not "") for each individual text box.

I think that's it, you gotta play with it.

"Jock" wrote:

Actually, thinking about this even further, how easy would it be to have the offending (empty) textbox(es) coloured yellow to make it obvious which one(s) need filling?

Might not be possible, but it would be impressive if it was.

—  
tia

Jock

"Steve the large" wrote:

You need to put a loop around the code with a flag that's set to the condition you want.  
for example

```
Private Sub CommandButton1_Click()  
Dim TB As Control  
Dim bAllFilled as boolean
```

```
bAllFilled = True 'Assume they are all  
filled in  
,
```

```
'Now loop through all text boxes to see if  
ANY are not filled in.  
,
```

```
For Each TB In UserForm1.Controls  
If TB.Name Like "TextBox*" Then  
If TB.Value = "" Then  
bAllFilled= False 'This line is executed  
if ANY are not filled  
End If  
End If  
Next  
,
```

'The following if..then displays the error msg

RE: Message boxes

```
only once, and aborts the
'commandbutton click event to give them a
chance to go back and fill
'in all six.
'
if not bAllFilled then
msgbox Prompt:="You have to fill in all
six!"
exit sub
end if
'
'your code reaches this point ONLY if all six
are filled in
'
<rest of handling of commandbutton1 click
event goes here.>

End Sub
```

"Jock" wrote:

Excellent job guys. How do  
I halt the code then until all  
boxes have been  
populated and the command  
button is clicked once again  
by the user?

--

tia

Jock

"Mike H" wrote:

Jock,

This will  
loop  
through all  
text boxes  
on a user  
form and  
generate a  
message

RE: Message boxes

for each  
unpopulated  
one. Instead  
of the  
message  
you could  
cancel data  
transfer to  
your  
worksheet.

```
Private Sub  
  CommandButton1_Click()  
  Dim TB As  
  Control  
  For Each  
    TB In  
      UserForm1.Controls  
    If TB.Name  
      Like  
        "TextBox*"  
    Then  
      If TB.Value  
        = "" Then  
        MsgBox  
          (TB.Name  
            & " isn't  
            populated")  
      End If  
    End If  
  Next  
End Sub
```

Mike  
"Jock"  
wrote:

I  
have  
six  
text  
boxes  
on  
a  
user  
form.  
How  
can  
I

RE: Message boxes

make  
a  
message  
box  
pop  
up  
if  
any  
of  
them  
haven't  
been  
populated  
when  
a  
command  
button  
is  
clicked  
to  
move  
the  
data  
to  
the  
spreadsheet?  
--  
tia  
  
Jock