

RE: Email and Fax a Report

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.modulesdaovba/2008-08/msg00061.html>

- *From:* dch3 <dch3@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 1 Aug 2008 18:15:01 -0700
-

And once again the need for an ACCESS-OUTLOOK INTEGRATION subgroup raises its head...please vote for it...

<http://www.microsoft.com/office/community/en-us/default.aspx?dg=microsoft.public.access.modulesdaovba&tid=cd>

Now to the response....

- 1) Its odd that you would have it in the Report_Open event. It will only delay the report opening as the code executes. If you spin it out to a Module, you'll be better off in the long run as there's then the potential to reuse it
- 2.) You may want to use a more descriptive label for the Error Label. I always go with the tried and true Err_[Procedure Name] same thing for the Exit label. But that's just me.
- 3) You're SETting a variable named 're' but referencing 'rs'. Shame, shame. I would recommend including OPTION EXPLICIT in all of your modules as it will catch typos like that. (DIM rs as DAO.Recordset BTW).
- 4) (.EOF Or .BOF) doesn't make any sense if your using .MoveNext in a DO UNTIL since it will be true on the first record and then keep looping till ..EOF. Personally I'm in the habit of using

```
While Not rs.EOF  
wend
```

- 5) Opening a report in a loop seems odd, especially from another report. What does. What does 'acknowledgment2' actually do? To me it would make more sense to open it, print it or print to your FAX. Since I don't know how you're faxing things out, I can't elaborate further. However, when it comes to sending an email, try this in its own procedure. Its something that I posted for another user that needed to send an email along with attachments. If you're not sending attachments, just delete the code that i've marked with 'ATTCH.

In terms of structuring your code to loop through the records, you've got the baisc loop down, it'd just be a matter of passing the relevant

RE: Email and Fax a Report

information to the sub below. By placing the sub on its own, it will allow you to reuse it again and again in any number of circumstances such as sending a manual confirmation if somebody calls in requesting on or if you're loop craps out.

Speaking of which, if I were sending emails in batch, I'd be inclined to having something in place that captures the Date/Time the email was sent and whether or not the code crapped out. That way you'll know if an email wasn't generated and can resend it if needed. Of course, you won't actually know if the email was delivered or crapped out due to a bad address, server problems, or whatnot. (But you might want to set up a specific email account to use as the .FROM so that any crap out messages are sent to it. I think that there's a way that you can specify a REPLY address so that when the user does REPLY the response doesn't go to the .FROM address. (That's a question for the Outlook newsgroup or Google it.)

By the way, did I mention that I really believe that there should be a Access-Outlook Integration Subgroup for questions like this?

```
sub createOutlookMailItem
```

```
Dim objOutlook As Outlook.Application  
Dim newMail As MailItem
```

```
Set objOutlook = CreateObject("Outlook.application")  
Set newMail = objOutlook.CreateItem(olMailItem)
```

```
newMail.To = "Bill.Gates@xxxxxxxxxxxxxx"  
newMail.Body = "The Office ribbon sucks like Dvorak"
```

```
'ATTCH Start  
Set newMailAttachments = newMail.Attachments
```

```
'If you're pulling records from a file, you'd start the loop here  
changing the next  
'line to file1 = rs("fieldName") or whichever way you prefer to pull the  
value from  
'the recordset you're working with, this code was done as a  
proof-of-concept,  
'so I used the file1 variable just to make the next line easier to read  
it can  
'be eliminated if needed  
'I would definitely check to see if the file exists first using the  
FileExists property  
'of the File System Object  
'Set fs = CreateObject("Scripting.FileSystemObject")  
'fs.FileExists(fileName)  
'If FileExists fails, you could then present the CommonDialog to give  
the users  
'the option of searching for the file themselves (see mvps.org for that  
aspect of
```

RE: Email and Fax a Report

RE: Email and Fax a Report

'it)

```
file1 = strTargetFolder & "\rptConfirmation.snp"  
newMailAttachments.Add file1, olByValue, 1
```

```
'end the loop here  
'ATTCH end
```

```
newMail.Subject = "Documents Confirmation"  
newMail.OriginatorDeliveryReportRequested = True  
newMail.ReadReceiptRequested = True  
newMail.Send
```

```
'ATTCH Set newMailAttachments = Nothing  
Set newMail = Nothing  
Set objOutlook = Nothing
```

end sub

You may also want to check out the `.SaveSentMessageFolder` property as it would allow you to have the sent messages saved in a specific folder – either at the user level or in a public folder.

[http://msdn.microsoft.com/en-us/library/aa171932\(office.11\).aspx](http://msdn.microsoft.com/en-us/library/aa171932(office.11).aspx)

"Jacinda" wrote:

I have asked a similar question before, but my previous inquiry was dealing with fax only... Here is what I'm trying to do:

We would like to electronically send acknowledgements for our orders, either by e-mail or by fax depending on the customer information (If e-mail is null then fax) Okay, so right now I have a button that will do that on an individual basis depending upon which client I pull up.

What I would like to do is a batch of acknowledgements...I'm willing to just work with e-mail for right now and print the one's with a fax but I'm hitting an error.–

Here is my code:

```
Private Sub Report_Open(Cancel As Integer)
```

```
On Error GoTo ProcError
```

```
Dim db As DAO.Database  
Dim rs As DAO.Recordset
```

RE: Email and Fax a Report

```
Set db = CurrentDb()
Set re = db.OpenRecordset("QAcknowledgement", dbOpenSnapshot)
With rs
Do Until (.EOF Or .BOF) = True
DoCmd.OpenReport "racknowledgment2", acViewNormal, _
"custno = " & rs("custno")
rs.MoveNext
Loop
End With

ExitProc:
'cleanup
On Error Resume Next
rs.Close: Set rs = Nothing
db.Close: Set db = Nothing
Exit Sub

ProcError
MsgBox "Error" & Err.Number & ": " & Err.Description, _
vbCritical, "Error in Test subroutine..."
Resume ExitProc

End Sub
```

The error I get is "compile error: sub or function not defined"

Jacinda