

By using this site you agree to the use of cookies for analytics, personalized content and ads.

[Learn More](#)



Announcements: 1

Question

[Applies to Office](#) | [Outlook](#) | [Microsoft Office Programming](#) | [Office 2013](#)

1018 views

Problem in VBA reading text from body of out of office messages in Outlook 2013 - Solved - see Reply 3

G G_in_Sydney asked on April 28, 2014 ✓

Hi

I'm not an expert in VBA, so this may be something obvious and I've missed it. About a month ago, I wrote a VBA program in an Excel 2013 workbook to go to a specified folder in my Outlook (the Inbox for one of my accounts) and search through all emails from a given date to look for any returned emails and, for those emails, to extract the email addresses contained in the text of the undelivered email. I put it together from code I found on the net, and eventually got it working fine.

The first pass searched for emails with "undeliverable" in the Subject and, for those emails, it then searched the Body of the text for emails and extracted them to the excel workbook. It now doesn't work. The initial selection of emails with "undeliverable" in the Subject line works, but somehow the characters on the .Body of the ReportItem (I've found out it's not a MailItem, as its returned mail) don't appear to copy to my string correctly, and, when I use the .Display function to show the email, the header and subject line are fine, but the text of the email appears to be Chinese characters.

However, when I look at that email (and the other returned emails) using Outlook 2013, the text in

the body is perfectly readable. Similarly using MsgBox to display the string into which I've copied the body of the email just shows lines of ????? . As a result, although it does find undelivered emails, it doesn't find any email addresses in the bodies of these emails

Extracts from my code are below. I appreciate any thoughts anyone has.

Thanks

G_in_Sydney

```
Dim MyOutLookApp As Object
Dim MyNameSpace As Object
Dim MyFolder As Object
Dim i, j, Index, index1, p, l, m, n As Integer
Dim itms, filteredItms As Outlook.Items
Dim MyDate As Variant
Dim olMail As Object
Dim EmailBody As String
Dim Check1 As Variant
Dim Start, EndofTexttoSearch, EmailAddressCounter, UndeliveredEmailAddresses As Integer
Dim TextatendofSearch As String
```

etc involving the workbook

etc

```
Set MyOutLookApp = CreateObject("Outlook.Application")
```

```
On Error GoTo ErrFlder
If MyOutLookApp Is Nothing Then
MsgBox "Please open Outlook first,then try again."
Exit Sub
End If
```

```
' Set MyNameSpace to contain all the email account folder names in Outlook
```

```
Set MyNameSpace = MyOutLookApp.GetNamespace("MAPI")
```

```
' Set MyFolder to folder number(15), then to the Inbox number (2)
Set MyFolder = MyNameSpace.Folders(15).Folders(2)
```

```
' Set itms to be the collection of all Outlook items within MyFolder
Set itms = MyFolder.Items
```

```
' Set filteredItms to filter items in MyFolder to on or after the date set above
Set filteredItms = itms.Restrict("[ReceivedTime]> " & Format(MyDate, "dddddd h:nn AMPM") & "")
MsgBox "filteredItms = " & filteredItms.Count ' I use to check it is selecting only after the set date
```

EmailAddressCounter = 1

UndeliveredEmailAddresses = 0

For Each olMail In filteredItems

If InStr(olMail.Subject, "Undeliverable") <> 0 Then

EmailBody = olMail.Body

Msgbox EmailBody ' Displays as lines of ???????

MsgBox Len(EmailBody) ' Displays positive numbers, typically around 2000 characters

olMail.Display ' Shows the body as Chinese characters, although the header of the email is

fine

Start = 1

etc etc to search through the text for email addresses

,

4 people had this question



Me Too

Reply



All replies (11)



Andreas Killer replied on April 28, 2014 ✓



MCC: Content Creator

Msgbox EmailBody ' Displays as lines of ???????

olMail.Display ' Shows the body as Chinese characters, although the header of the email is fine

a) Msgbox EmailBody

That is normal. The string EmailBody contains Unicode chars and the MsgBox is not able to show them correctly.

b) olMail.Display

That is also normal, the mail is written in Chinese.

Andreas.

Be the first person to mark this helpful



 Helpful

Reply

**G** G_in_Sydney replied on April 28, 2014 ✓ In reply to Andreas Killer's post on April 28, 2014

Hi Andreas

Thanks for the reply. A couple of follow ups:

- 1.The undelivered messages are not in Chinese. They all come from a local company here in Australia and read perfectly normally in English when I open them using Outlook 2013, so I presume that the actual text is not Unicode.
2. Your observation that the VBA code/Outlook/Office settings I have seems to be interpreting them as Unicode looks correct. Any ideas why this should be and what I need to change to get it to read as a normal string?

Thanks

G_in_Sydney

Be the first person to mark this helpful

 Helpful

Reply

**G** G_in_Sydney replied on April 29, 2014 ✓ In reply to G_in_Sydney's post on April 28, 2014

To anyone that has the same problem

I've managed to develop a work around to solve the problem (interestingly my googling found a reference to the problem of the ReportItem.Body not returning a readable string back in 2000, and referred to it a a known bug then!). I've changed my code to use the ReportItem.SaveAs property to save the entire message as a Text file, open the text file and read that into my string, and then 'top and tail' it to delete the header and footer parts I don't want to search. The email search code then works fine on this string and produces the bounced email addresses. The relevant code is below.

Happy coding

G_in_Sydney

```
Dim MyOutLookApp As Object
Dim MyNameSpace As Object
Dim MyFolder As Object
Dim Click1 As Variant
Dim i, j, Index, index1, p, l, m, n As Integer
Dim itms, filteredItms As Outlook.Items
Dim MyDate As Variant
Dim olMail As Object
Dim olMailItem As ReportItem
Dim EmailBody As String
Dim Check1 As Variant
Dim MessageSaveType As Variant
Dim Start, EndofTexttoSearch, EmailAddressCounter As Integer
Dim TextatendofSearch, CheckStr, Charlist, Outstr, GetStr, TextatStartofSearch As String
```

etc code working with workbook, then

```
' Set MyNameSpace to contain all the email account folder names in Outlook
'
'   Set MyNameSpace = MyOutLookApp.GetNamespaces("MAPI")
'
' Set MyFolder to folder number(15), and the Inbox number (2), the folder I want to search
Set MyFolder = MyNameSpace.Folders(15).Folders(2)
'
' Set itms to be the collection of all Outlook items within MyFolder
Set itms = MyFolder.Items   ' MsgBox "Number of items in MyFolder = " & itms.Count
'
' Set filteredItms to filter items in MyFolder to on or after the date set above
Set filteredItms = itms.Restrict("[ReceivedTime]> " & Format(MyDate, "dddd h:nn AMPM") & "")
'MsgBox "filteredItms = " & filteredItms.Count
'
'
'   EmailAddressCounter = 1

For Each olMail In filteredItms
    If olMail.MessageClass = "REPORT.IPM.Note.NDR" Then ' Message class for Returned Emails
        ' save the message as a text file in a temporary folder, as cannot directly read the body of a
        ReportItem
        Set olMailItem = olMail
        MessageSaveType = olTXT
        Tempfilepath = "C:\changethefoldertosuityourneeds\TempDoNotDelete\TempFile.txt" ' note:
        this does not check the directory exists, so create it first
```

```

    olMailItem.SaveAs Path:=Tempfilepath, Type:=MessageSaveType
' Open temp text file and read the text into the EmailBody string
iFile = FreeFile
Open Tempfilepath For Input As #iFile
EmailBody = Input$(LOF(iFile), iFile)
Close iFile
' Delete TempFilePath
Kill Tempfilepath
,
' Delete the header down to the Subject line to avoid *** Email address is removed for privacy ***
and duplicate versions of the email addresses
TextatStartofSearch = "Subject:"
TextatendofSearch = "Diagnostic information for administrators"
EmailBody = Right(EmailBody, Len(EmailBody) - InStr(1, EmailBody, TextatStartofSearch) + 1)
Start = 1
,
' Delete the tail of the email body which isn't needed to be searched for email addresses
EndofTexttoSearch = InStr(Start, EmailBody, TextatendofSearch)
If EndofTexttoSearch = 0 Then 'if email message doesn't contain this text, default to the
full string
EndofTexttoSearch = Len(EmailBody)
End If
EmailBody = Left(EmailBody, EndofTexttoSearch - 1)
,
' continue with the code to search the string for email addreses etc etc

```

5 people found this helpful

 Helpful

Reply

...

GM GMBrook replied on October 8, 2014 ✓

 In reply to G_in_Sydney's post on April 29, 2014

G_in_Sydney - great solution!

I had the same problem and your code has given me the work around.. and no more 'Chinese' emails.

Many thanks for sharing.

Be the first person to mark this helpful

 Helpful

Reply



G G_in_Sydney replied on October 9, 2014 ✓

 In reply to GMBrook's post on October 8, 2014

GMBrook - glad to help. I've used other people's solutions so many times, it's good to be able to contribute back to the community. Regards G_in_Sydney

Be the first person to mark this helpful

 Helpful

Reply



CR CrisCasey replied on January 22, 2015 ✓

 In reply to G_in_Sydney's post on October 9, 2014

G - I find myself in need of a similar tool and have just started my search. Is it possible to get a full copy of your code?

Thanks and kindest regards.

1 person found this helpful

 Helpful

Reply



JoePaulsson replied on February 19, 2015 ✓

↳ In reply to CrisCasey's post on January 22, 2015

Hi,

The write to file and then read from file solution works. But I didn't like it. I don't want the extratime and also many thousands of read/write operations to harddrive. Not good if you have a SSD that are not made for many write operations. After investigating the whole very annoying thing I came to the following solution:

```
private string ConvertToCorrectFormat(Outlook.ReportItem ri) {  
    string str = ri.Body;  
    Encoding enc = Encoding.Unicode;  
    byte[] bBEArr = enc.GetBytes(str);  
    if (bBEArr[1] == 0 && bBEArr[3] == 0) //If OK string -> return str before converting  
        return str;  
    int sizeOfNewArr = str.Length * 4;  
    byte[] arrNew = new byte[sizeOfNewArr];  
  
    for (int i = 0; i < bBEArr.Length; i += 2) {  
        byte b1 = bBEArr[i];  
        byte b2 = bBEArr[i + 1];  
  
        arrNew[i * 2] = b1;  
        arrNew[i * 2 + 2] = b2;  
    }  
    str = enc.GetString(arrNew);  
    return str;  
}
```

/Joe Fjell

<http://www.nfi.se/kurs/c-sharp.aspx>

You and 1 other person found this helpful

Helpful

Reply



G G_in_Sydney replied on February 20, 2015 ✓

↳ In reply to JoePaulsson's post on February 19, 2015

Hi Joe

Thanks for posting your approach. I'm afraid it's beyond my experience, so I'm not sure how it works, but I'm sure other people will appreciate an alternative without creating a temporary file.

Regards G_in_Sydney

Be the first person to mark this helpful

▲
Helpful

Reply

...

BL Bluey47 replied on April 14, 2015 ✓

Just had this happen to me. Used StrConv to convert back to Unicode.

```
Dim strBody as String  
strBody = StrConv(olMail.Body,vbUnicode)
```

3 people found this helpful

▲
Helpful

Reply

...

TO toha-toha replied on September 16, 2016 ✓

↳ In reply to JoePaulsson's post on February 19, 2015

```
The recipient's mailbox is full and can't accept  
messages now. Microsoft Exchange will not try to  
redeliver this message for you. Please try  
resending this message later, or ☹ @-
```

Dear Mr. Joe Paulsson,

I try your suggestion.

My app written in VB.net

I got problem at getting email header using your code, like the Symbols character at the end of result.

So far, this is what I have tried (3 ways - 2 last commented):

//After convert email body with your code snippet

```
sEmailBody = Regex.Replace(sEmailBody, "[^A-Za-z0-9\-\@\s\n]", "")
```

```
'sEmailBody = Console.BEARUtility.RemoveSymbol(sEmailBody)
```

```
'sEmailBody = sEmailBody.Substring(0, sEmailBody.Length - 3)
```

After using Regex, the email header appears on string.

I think it is failure on conversion the email header below NDR email.

Like this:

Therrecipientse-mailaddresswasnotfoundintherecipientse-mailsystem.MicrosoftExchangewillnottrytoredeliverthismessageforyou.Pleasecheckthe-mailaddressSdyresendingO::aorprovidethefollowingdiagnostictexttoyourssystemadministrator.ThefollowingorganizationrejectedyourAOe:3R:4.us.message:labs.com.rrrrrDXSNcinformationforIR:FN@:Generatingserver::8F01.aia.bizA:N.AQz@F.t.co.idcluster4.us.message:labs.comlt:CC5r4gs.message:labs.com5.1.1SMTP550Ia:ltA:N.AQz@F.t.co.idgt5.1.1gtSMTPOriginalmessageheaders:S89h:ltID.pn.estatement@@a.comgtReceived:3Fmpps.filterd:8F01127.0.0.1bymycspxsmt01.aia.biz8.14.5/8.14.5withSMTPidt0SE38mB011718forltA:N.AQz@F.t.co.idgtYDM28Jan201522:05:440800Received:Nf:8A02X10.50.163.135by5bpxsmt01.aia.bizwithESMdn6t1-1forltA:N.AQz@F.t.co.idgtYDM28Jan201522:05:440800Received:Nf:8A02X10.50.163.135bymycspwdss02.aia.bizwithMBHF4eVC6.0.3790.3959Wed28Jan201522:06:350800Received:fromMYCSPWEDG02.AIA.Biz10.50.163.130bymycspwdss02XChLtSMTPSVC6.0.3790.3959Wed28Jan201522:06:330800Received:NfMYCWHUB01.AIA.BIZ10.50.162.73byMYCWEDG02.AIA.Biz10.50.163.130withLtSMr6c8.3.348.2Wed28Jan201522:06:490800Received:fromdafp3aiarms0110.49.18.65byMYCSPWHUB01.AIA.BIZ10.50.162.73withMBHF4Pdd8.3.348.2Wed28Jan201522:06:330800Received:NfCGKDCPWPFD0310.49.18.7bydafp3aiarms01withMBHFSMTPSVC8.0.9200.16384Wed28Jan201521:06:330700MIME-Version:1.0From:ltID.pn.estatement@@a.comgtTo:ltA:N.AQz@F.t.co.idgtDate:L28Jan201521:06:330700Subject:LSDSaPesertaDPLKAIAFSS::G013RincianLS-gMContent-Type::NN:boundaryquot--boundary676659979f-32f9-47f8-a735-7307adc3323aquotMessage-ID:ltDAFP3AIARMS01JdnjQ000432b1@dafp3aiarms0184tX-OriginalArrivalTime:28Jan201514:06:33.0037UTCFILETIME9EE31FD0:01D03B03X-Proofpoint-Virus-Version:LSreSD2.50.10432:5.13.681.0.330.0.0000definitions2015-01-2801:2015-01-282015-01-281970-01-01signatures0X-Proofpoint-Spam-D@:S:u:KBHyW:Nt5e:5e0suspectscore3phishscore0adultscore0bulkscore0classifier:EGCtS3eTA1engine7.0.1-1402240000definitionsmain-1501280146/html

(the regex is not corrected so, the result is broken like that after using the regex)

Be the first person to mark this helpful

▲
Helpful

Reply



2 [Next](#) >

Still have questions?


[Ask the community](#)

[Ask Microsoft virtual agent](#)

Related Content

[Show more content like this](#)



 English

[Microsoft Community Code of Conduct](#)

[Community Participation Center](#)

[Trademarks](#)

[Privacy & Cookies](#)

[Terms of Use](#)

Microsoft © 2016 Microsoft

[Enable](#)

Enable UC auto page to scroll all the way down through pages!