Send Report in an Email

October
2024

CROW
WAREZ
www.sosa.tv

Excel VBA | Send Report in an Email

Summary

In this tutorial, we will create a macro to send a report using Visual Basic for applications, along with Excel and Outlook.

This becomes very convenient when a report needs to be sent regularly in a consistent way to a specific group of recipients, with a file that can be updated within Excel, without opening a previously sent email and changing the information, which is prone to errors if it has to be delivered right away with little time to spare.

We will add a table in the email, and also the Excel file attached.

For this tutorial, we will be using Windows, as at this moment the macro has not been developed for Mac.

Visit www.sosa.tv for information about some data analytics projects I've worked on and other cool stuff.

Youtube video here.

About Working With Excel Macros In Tandem with Outlook

We need to add a reference in order to have Word to be the HTML editor of our email.

To accomplish this, we need to open the Visual Basic Editor, then go to the menu Tools > References scroll down, find and check Microsoft Word 16.0 Object Library.

We created a finished report available on sosa.tv, called:

Email.xlsb

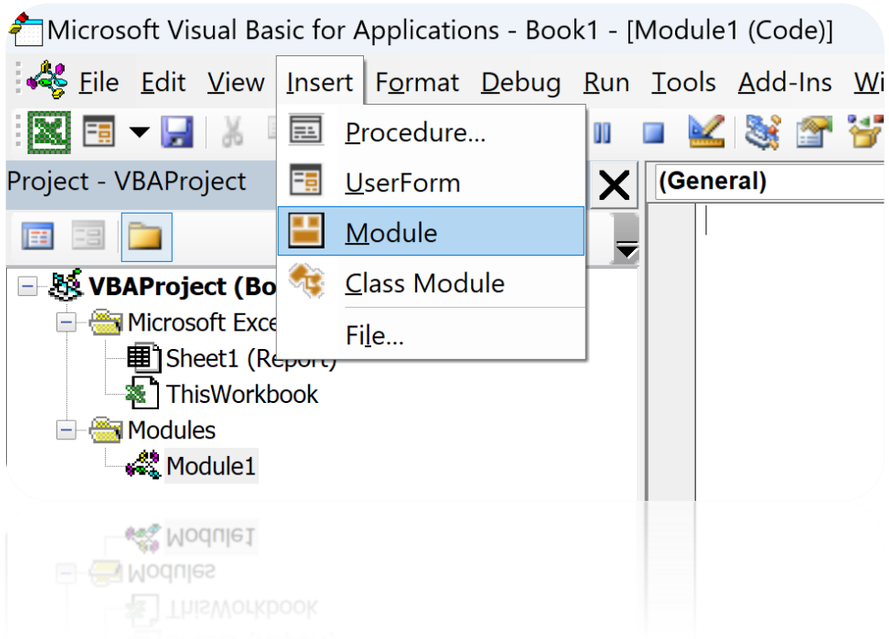
Download the working macro file here

Instructions to Create the file from scratch

You can achieve this by following these steps:

Step 1. Open Visual Basic by pressing alt + F8 or going to the Developer menu if visible, and clicking on Visual Basic.

Step 2. Add a new module if there is not one already.



Step 3. Paste the code below.

But first, about this code:

The Sub Mail_Workbook_Outlook contains the macro that will gather and prepare the objects and data to be added into the email.

While the Function RangetoHTML(rng as Range), will get the table from a range in an Excel spreadsheet as a table and convert this data into a HTML table. This is, in my experience, that recipients prefer to have the data visible in the body of the email for a first impression, in addition to the attached file where they can filter or add formulae to reconciliate and check using their own calculations. Note how the range is declared inside the function: rng as Range.

The code to paste:

```
Sub Mail_workbook_Outlook()

    'Working in Excel 2000-2016
    'This example sends the last saved version of the Activeworkbook

    Dim sRecipients As String   '   stores recipients list
    Dim sCC As String           '   stores CC list
    Dim sSubject As String      '   stores subject
    Dim rngTable As Range

    Dim wdDoc As Word.Document

    Dim OutApp As Object
    Dim OutMail As Object
    Dim oTestOutApp As Object

    Set OutApp = CreateObject("Outlook.Application")
    Set OutMail = OutApp.CreateItem(0)

    '   checki if outlook is , if not, open outlook
```

```
On Error Resume Next
    Set oTestOutApp = GetObject(, "Outlook.Application")
On Error GoTo 0

If oTestOutApp Is Nothing Then
    Shell ("OUTLOOK")
End If

'   set word as html editor
Set wdDoc = OutMail.GetInspector.WordEditor

' get table range, here we use a named range called "Stock_Report_Table"
Set rngTable = ThisWorkbook.Sheets("Report").Range("Stock_Report_Table")

'   Text is in HTML
sBody = "<font size=""11pt"" face=""Calibri"" color=""black"">" _
    & "Hello," & "<br>" _
    & "<br>" _
    & "Please see attached for a summary of this report." & "<br>" _
    & "</font>" & RangetoHTML(rngTable)

'   <br> is for line break
sSignature = "<font size=8pt face=""Helvetica"" color=""red"">" _
    & "<br>" _
    & "<br>Thank you,<br>" _
    & "<b>CrowWarez</b><br>" _
```

```
    & "<a href=""https://www.instagram.com/antoniososa7997/"">Instagram</a> | " _
    & "<a href=""https://x.com/a_tsosa"">X</a> | " _
    & "<a href=""https://www.facebook.com/antonio.sosa"">Facebook</a>/ | " _
    & "<a href=""https://www.youtube.com/@CrowWarez-l3m"">Youtube</a></font>"


'   Edit recipients list or use a range in a worksheet
sRecipients = _
"Person who receives the report <email@domain.net>; "

sCC = "email2@domain.net"

'   subject has a dynamic date
sSubject = "Sample Report - " & Format(Now, "Mmmm dd yyyy")

On Error Resume Next
With OutMail
    .To = sRecipients
    .CC = sCC
    .BCC = ""
    .Subject = sSubject
    .HTMLBody = sBody & sSignature
        'You can add other files also like this
        .Attachments.Add ActiveWorkbook.FullName
    .Attachments.Add sFilename
    .Attachments.Add sPDF
```

```
        .Display ' or use Send
    End With
    On Error GoTo 0

    Set OutMail = Nothing
    Set OutApp = Nothing

End Sub

Function RangetoHTML(rng As Range)

    '   Changed by Ron de Bruin 28-Oct-2006
    '   Working in Office 2000-2016
    '   This function converts an Excel table into HTML

    Dim fso As Object
    Dim ts As Object
    Dim TempFile As String
    Dim TempWB As Workbook

    TempFile = Environ$("temp") & "\" & Format(Now, "dd-mm-yy h-mm-ss") & ".htm"

    '   Copy the range and create a new workbook to past the data in
    rng.Copy
    Set TempWB = Workbooks.Add(1)
    With TempWB.Sheets(1)
```

```
    .Cells(1).PasteSpecial Paste:=8
    .Cells(1).PasteSpecial xlPasteValues, , False, False
    .Cells(1).PasteSpecial xlPasteFormats, , False, False
    .Cells(1).Select
    Application.CutCopyMode = False
    On Error Resume Next
    .DrawingObjects.Visible = True
    .DrawingObjects.Delete
    On Error GoTo 0
End With

'   Publish the sheet to a htm file
With TempWB.PublishObjects.Add( _
     SourceType:=xlSourceRange, _
     Filename:=TempFile, _
     Sheet:=TempWB.Sheets(1).Name, _
     Source:=TempWB.Sheets(1).UsedRange.Address, _
     HtmlType:=xlHtmlStatic)
    .Publish (True)
End With

'   Read all data from the htm file into RangetoHTML
Set fso = CreateObject("Scripting.FileSystemObject")
Set ts = fso.GetFile(TempFile).OpenAsTextStream(1, -2)
RangetoHTML = ts.readall
ts.Close
```

```
    RangetoHTML = Replace(RangetoHTML, "align=center x:publishsource=", _
                          "align=left x:publishsource=")

    '   Close TempWB
    TempWB.Close savechanges:=False

    '   Delete the htm file we used in this function
    Kill TempFile

    Set ts = Nothing
    Set fso = Nothing
    Set TempWB = Nothing

End Function
```

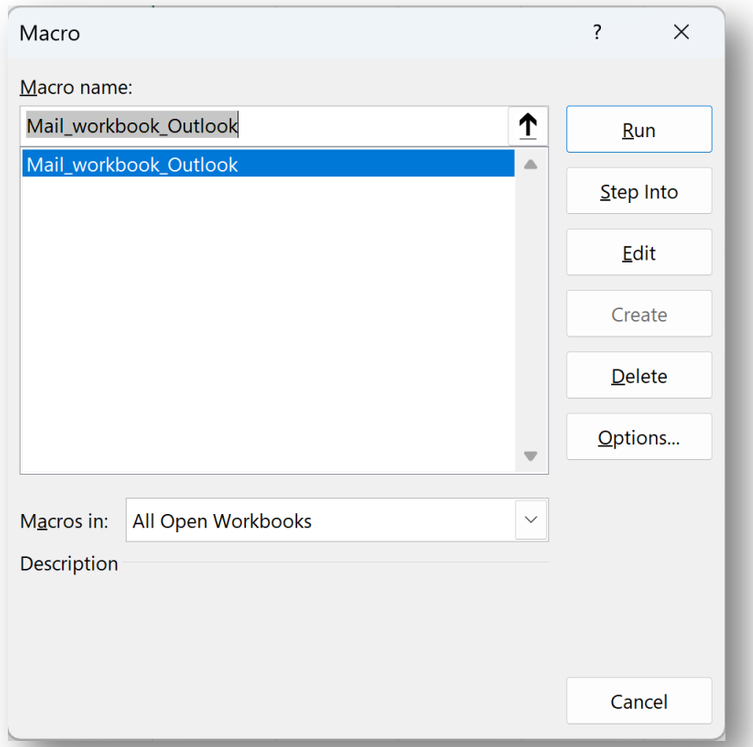
Step 4. Close Visual Basic, click on the X at the top right or the Excel logo at the left, below the menu.

Step 5. In Excel, rename the spreadsheet *"Sheet1"* to *"Report"*, then on cell A1, enter the sample table below:

Sample Report			
Account	Account Description	Stock Units	Stock Price
10005	J Peterman	1000	$ 121.00
40120	Kramerica Industries	575	$ 75.00
61015	Vandelay Imports / Exports	14	$ 3.00
55500	Hennigan's Whisky	457	$ 55.00
40110	Kruger Industrial Smoothing	671	$ 4.00
77150	Pendant Publishing	77	$ 98.00

Then, **select cells A1 to D12** and create a named range by clicking on the **Name Box**, just above the intersection of the Columns and Rows headers and with **A1:D12** still selected, enter the name **Stock_Report_Table**.

Step 6. Press F5 to open the Run Macro window *(Image to the right)*.
Step 7. Select macro and click Run.



How the Macro and the Function Work

The Sub Mail_workbook_Outlook() does the following tasks:

1) Declare all the variables and objects we will be using in our process. Examples of these are:
sRecipients, which is a text variable we will use to store the email addresses we will be sending our report, remember that multiple email addresses need to be separated by a semicolon **(;)**.
OutApp is an object that we will set as CreateObject("Outlook.Application"). This object needs to be referenced in VBA, by going in VBA to Tools > References scrolling down and finding and checking Microsoft Word 16.0 Object Library.
We add Word as it is needed as the HTML editor to produce the email just the way we want it.

2) Set the objects and variables.
If these are not set, the macro will crash as there is vital information that VBA needs in order to work. Examples:
Set OutApp = CreateObject("Outlook.Application"). OutApp is set as the Outlook application, this way VBA can create a new email, add a Subject, a Body, Recipients, etc.

We check if Outlook is already open, if not, we open it with the line: *'Shell ("OUTLOOK")'*:

```
On Error Resume Next
    Set oTestOutApp = GetObject(, "Outlook.Application")
On Error GoTo 0

If oTestOutApp Is Nothing Then
    Shell ("OUTLOOK")
End If
```

```
sRecipients = _
  "Person who receives the report <email@domain.net>; "
```

sRecipients is a string, or text string, you can update this in the VBA code or use a range in Excel with a list of emails, same for the body, CC, CCO.

3) With OutMail object:

Here is where we tell Outlook to finally get all the data to create the email:

```
With OutMail
    .To = sRecipients
    .CC = sCC
    .BCC = ""
    .Subject = sSubject
    .HTMLBody = sBody & sSignature
        'You can add other files also like this
        .Attachments.Add ActiveWorkbook.FullName
    .Attachments.Add sFilename
    .Attachments.Add sPDF
    .Display ' or use .Send
End With
```

Note that the next to last line we use .Display to have the new email opened before sending it. If you are confident enough about the data, you can replace it with .Send and OutMail will send it immediately.

The Function RangetoHTML(rng As Range) does this:

1) Declare all the variables and objects just as with the Sub.

2) Declare a temporary file name to create the HTML content we are getting from Excel format.

3) Process the table:

Copy the table.

Add a temporary workbook to process the table.

Paste the copied table, its formats and other attributes.

Publish the sheet to a htm file.

Read the data from that temporary htm file in order to be inserted into the email.

Close and delete temporary files.

There are many updates or improvements we can make to this macro file, such as adding a pdf attachment, create lists to store all the recipients or body or signature, add pictures, completely automate running the macro and sending the email in the background at a particular time of day, like before the work day starts or after hours.

If you need a specific update, send me an email to antonio@sosa.tv or leave a comment in the video and I will be happy to help.

Visit www.sosa.tv for information about some data analytics projects I've worked on and other cool stuff.