Email Generation

Created by:	Orhan Celiloglu	**Reviewed by:**	Marek Prazak	**Create Date:**	18 April 2019

1. Purpose

- This lesson will introduce you how to generate emails with Macro.
- The lesson should take approximately 1 hour to finish.

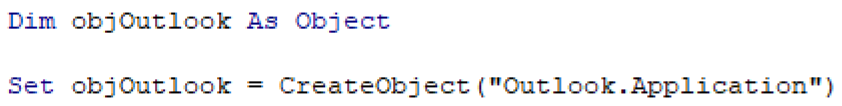
2. Definition

- Email generation is very manual task which can be easily automated. Adding attachments, applying special formatting, CC, BCC settings can all be automated.
- There are two very useful methods you will see in this training. First, using a template to generate emails and second how to loop a list of employees in order to send separated and unique emails.

3. Content

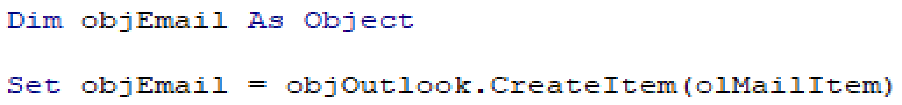
3.1 Email Generation

- First you have to set up Outlook Application
- Declare variable that will hold object and assign it outlook application



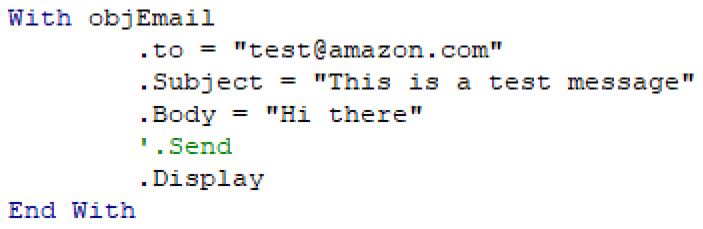
```
Dim objOutlook As Object

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

- Create Email Object by declaring variable of object type and assign item from outlook



```
Dim objEmail As Object

Set objEmail = objOutlook.CreateItem(olMailItem)
```

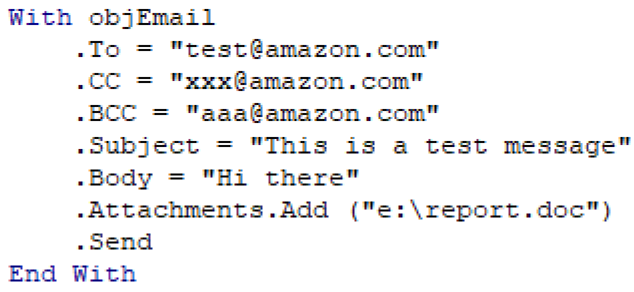
- Generating Email – We will use With/End with (explained in following lesson) as this is cleanest and fastest option for the code.



```
With objEmail
        .to = "test@amazon.com"
        .Subject = "This is a test message"
        .Body = "Hi there"
        '.Send
        .Display
End With
```

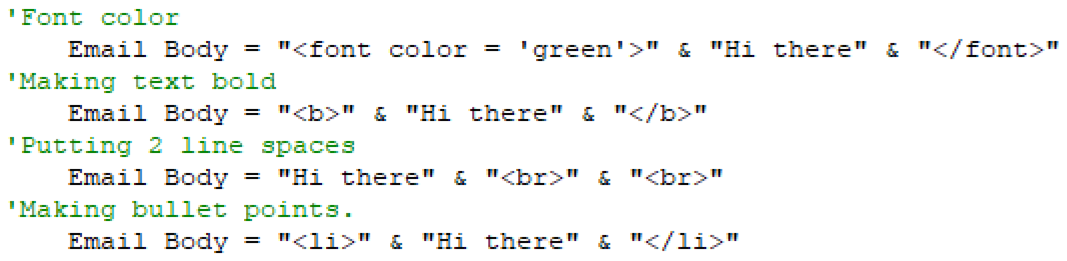
Note: *In this case we've left Send disabled and only display visible, this means that when you run the macro the email* ***only*** *opens.*

- You can also Add Attachment, CC and BCC



```
With objEmail
    .To = "test@amazon.com"
    .CC = "xxx@amazon.com"
    .BCC = "aaa@amazon.com"
    .Subject = "This is a test message"
    .Body = "Hi there"
    .Attachments.Add ("e:\report.doc")
    .Send
End With
```

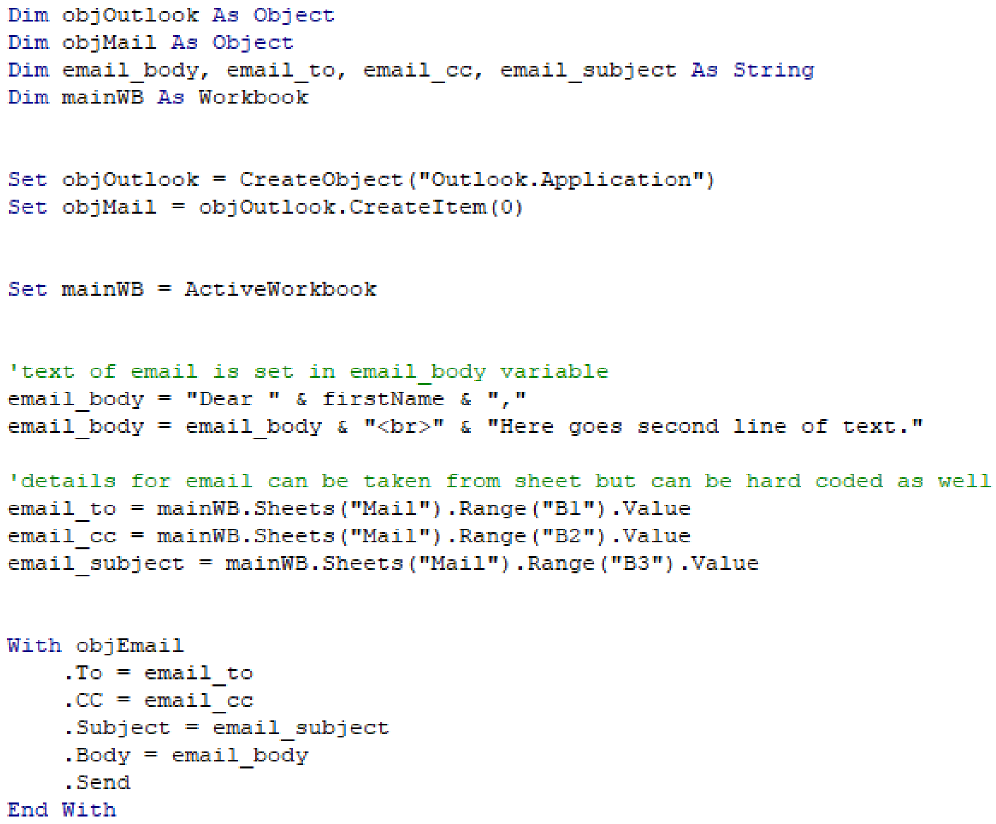
- Formatting of generated emails with HTML, if you're familiar with HTML this will be nothing new. For anyone is not familiar with HTML, do not worry. It is coding language mainly used for creation of web pages and we are utilizing few of its features to adjust the text of email. Logic of the syntax is easily understandable from below. If you want to read more into this language, we recommend link in additional knowledge chapter.



```
'Font color
    Email Body = "<font color = 'green'>" & "Hi there" & "</font>"
'Making text bold
    Email Body = "<b>" & "Hi there" & "</b>"
'Putting 2 line spaces
    Email Body = "Hi there" & "<br>" & "<br>"
'Making bullet points.
    Email Body = "<li>" & "Hi there" & "</li>"
```

3.1.1Generate email with Body in code

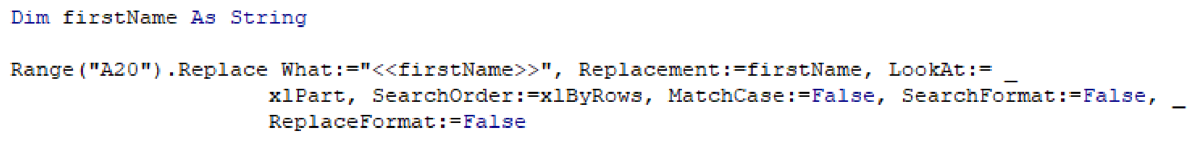
- In this case content of the email will be hardcoded in the VBA as per below syntax. Downside to this solution is that person without knowledge of VBA will not be able to update the template.
- Text, Dates, Formatting are handled through the code as per above HTML formatting.



```
Dim objOutlook As Object
Dim objMail As Object
Dim email_body, email_to, email_cc, email_subject As String
Dim mainWB As Workbook


Set objOutlook = CreateObject("Outlook.Application")
Set objMail = objOutlook.CreateItem(0)


Set mainWB = ActiveWorkbook


'text of email is set in email_body variable
email_body = "Dear " & firstName & ","
email_body = email_body & "<br>" & "Here goes second line of text."

'details for email can be taken from sheet but can be hard coded as well
email_to = mainWB.Sheets("Mail").Range("B1").Value
email_cc = mainWB.Sheets("Mail").Range("B2").Value
email_subject = mainWB.Sheets("Mail").Range("B3").Value


With objEmail
    .To = email_to
    .CC = email_cc
    .Subject = email_subject
    .Body = email_body
    .Send
End With
```

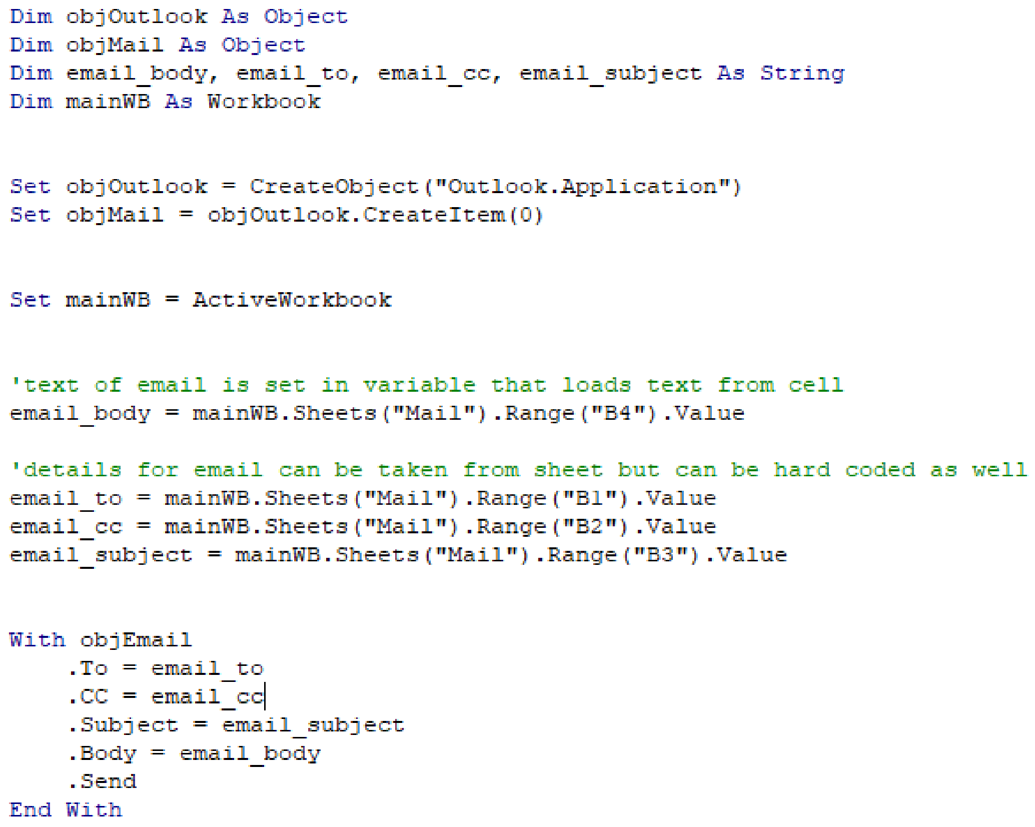
3.1.2Generate email with Body in Sheet

- Compared to hardcoded email this option offers even users without VBA knowledge option to update the email.
- Formatting and adjustment of the text is done in the cell.
- Changing text/names/dates can be done through use of replace function in the cell containing the text (below syntax).

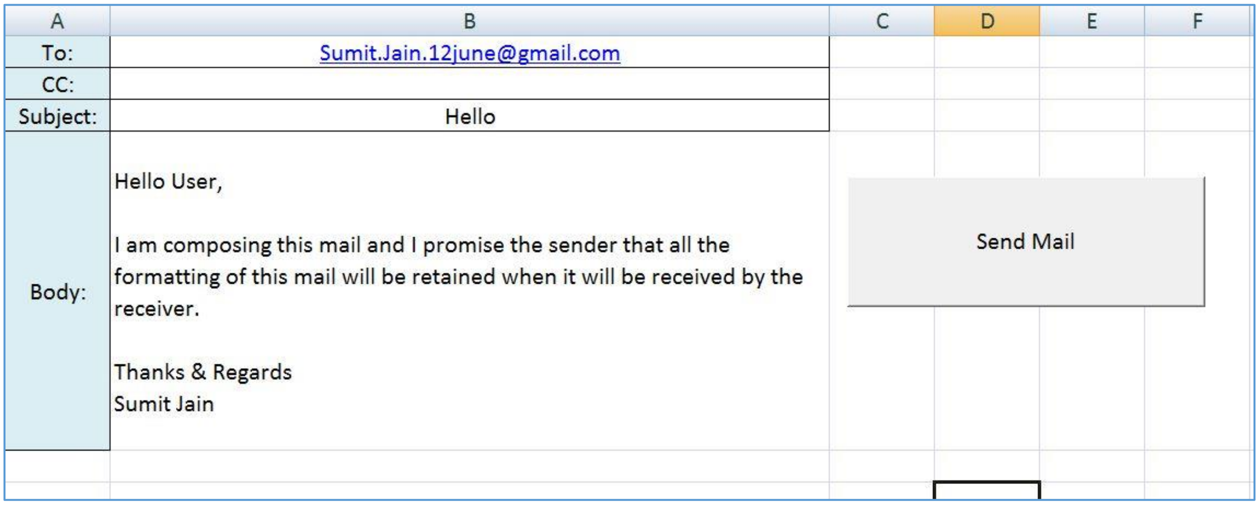


```
Dim firstName As String

Range("A20").Replace What:="<<firstName>>", Replacement:=firstName, LookAt:= _
                    xlPart, SearchOrder:=xlByRows, MatchCase:=False, SearchFormat:=False, _
                    ReplaceFormat:=False
```

- Email syntax



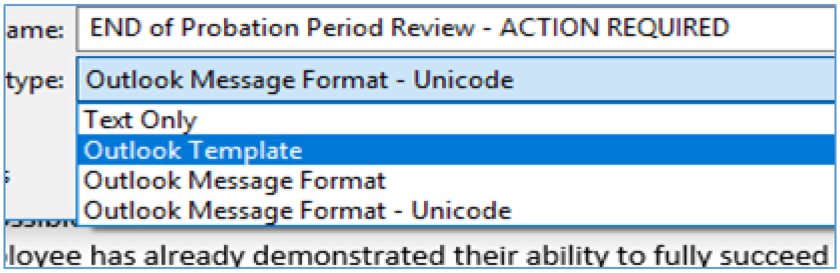
```
Dim objOutlook As Object
Dim objMail As Object
Dim email_body, email_to, email_cc, email_subject As String
Dim mainWB As Workbook


Set objOutlook = CreateObject("Outlook.Application")
Set objMail = objOutlook.CreateItem(0)


Set mainWB = ActiveWorkbook


'text of email is set in variable that loads text from cell
email_body = mainWB.Sheets("Mail").Range("B4").Value

'details for email can be taken from sheet but can be hard coded as well
email_to = mainWB.Sheets("Mail").Range("B1").Value
email_cc = mainWB.Sheets("Mail").Range("B2").Value
email_subject = mainWB.Sheets("Mail").Range("B3").Value


With objEmail
    .To = email_to
    .CC = email_cc
    .Subject = email_subject
    .Body = email_body
    .Send
End With
```



A	B	C	D	E	F
To:	Sumit.Jain.12june@gmail.com				
CC:					
Subject:	Hello				
Body:	Hello User, I am composing this mail and I promise the sender that all the formatting of this mail will be retained when it will be received by the receiver. Thanks & Regards Sumit Jain				

3.1.3Generate Email using Outlook template

- This is the most reasonable solution for creating the email generating macro, as it is easiest to maintain for regular users.
- The email template is created as a new email in Outlook and then saved as the .OFT type (Outlook Template).



ame: END of Probation Period Review - ACTION REQUIRED

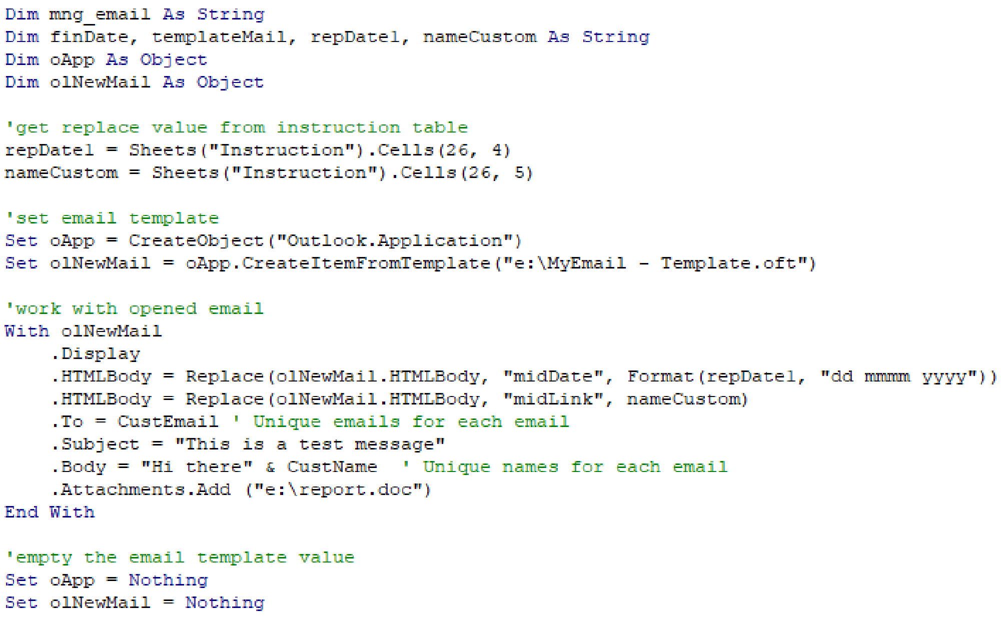
type: Outlook Message Format - Unicode

loyee has already demonstrated their ability to fully succeed

- Value that will be **changing** due to mail merge is handled by **function replace**

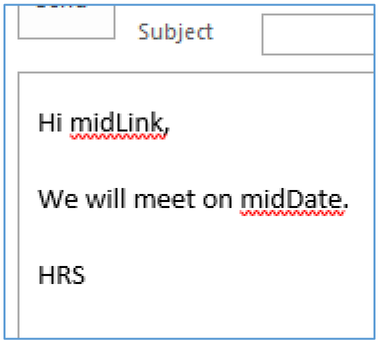
```
.HTMLBody = Replace(olNewMail.HTMLBody, "midDate", Format(repDate1, "dd mmmm yyyy"))
.HTMLBody = Replace(olNewMail.HTMLBody, "midLink", nameCustom)
```

  - **Replace(** WHERE**,** WHAT TO REPLACE**,** WITH WHAT REPLACE**)**
  - In **opened template** we want to replace the code word **midDate** with variable **repDate1** that contains date, which we are formatting into 24 April 2018.



```
Dim mng_email As String
Dim finDate, templateMail, repDate1, nameCustom As String
Dim oApp As Object
Dim olNewMail As Object

'get replace value from instruction table
repDate1 = Sheets("Instruction").Cells(26, 4)
nameCustom = Sheets("Instruction").Cells(26, 5)

'set email template
Set oApp = CreateObject("Outlook.Application")
Set olNewMail = oApp.CreateItemFromTemplate("e:\MyEmail - Template.oft")

'work with opened email
With olNewMail
    .Display
    .HTMLBody = Replace(olNewMail.HTMLBody, "midDate", Format(repDate1, "dd mmmm yyyy"))
    .HTMLBody = Replace(olNewMail.HTMLBody, "midLink", nameCustom)
    .To = CustEmail ' Unique emails for each email
    .Subject = "This is a test message"
    .Body = "Hi there" & CustName  ' Unique names for each email
    .Attachments.Add ("e:\report.doc")
End With

'empty the email template value
Set oApp = Nothing
Set olNewMail = Nothing
```

- The “code words” are used the same way as is done in the mail merge in Word.



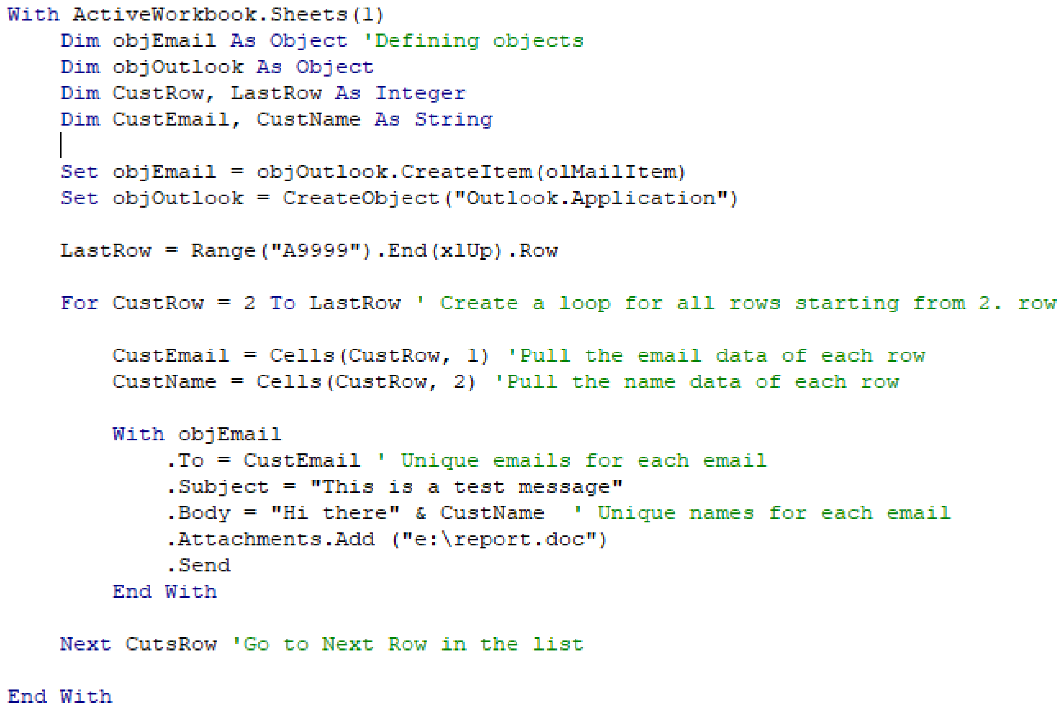
Hi midLink,

We will meet on midDate.

HRS

3.1.4 LOOP THROUGH EACH ROW AND SEND SEPARATE EMAILS TO LIST OF PEOPLE

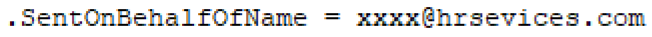
- In cases where you have to send out large amount of records, you can loop through the list that you’ve created (similar to previous lesson).



```
With ActiveWorkbook.Sheets(1)
    Dim objEmail As Object 'Defining objects
    Dim objOutlook As Object
    Dim CustRow, LastRow As Integer
    Dim CustEmail, CustName As String

    Set objEmail = objOutlook.CreateItem(olMailItem)
    Set objOutlook = CreateObject("Outlook.Application")

    LastRow = Range("A9999").End(xlUp).Row

    For CustRow = 2 To LastRow ' Create a loop for all rows starting from 2. row

        CustEmail = Cells(CustRow, 1) 'Pull the email data of each row
        CustName = Cells(CustRow, 2) 'Pull the name data of each row

        With objEmail
            .To = CustEmail ' Unique emails for each email
            .Subject = "This is a test message"
            .Body = "Hi there" & CustName  ' Unique names for each email
            .Attachments.Add ("e:\report.doc")
            .Send
        End With

    Next CutsRow 'Go to Next Row in the list

End With
```

3.2 TIPS AND TRICKS FOR EMAIL GENERATION

- For enabling Outlook activities in Macro, please go Tools tab in Visual Basic and from there open the References dialog box and find Microsoft Outlook 14.0 Object Library (higher version) and click the check box and press OK.
- It's also possible to send email from a Shared Email Box by using below code, that is to be added in the same section as CC and send to.



```
.SentOnBehalfOfName = xxxx@hrsevices.com
```

- If you don't use "SentOnBehalfOfName" code, the email will be sent automatically from your email address which is defined in outlook.

4. Closing Exercise

Complete below exercise.

1) In source excel you have three buttons. Create 3 modules and assign the macros to buttons.
    a. Create macro that will generate email from Hardcoded template.
        i. Email will look like source template
        ii. Send to Name will change in body of email and in subject according to source
        iii. Where to, when will change according to source
        iv. Send to name will be red format, weekend will be bold, where to/when will be yellow
        v. **Mail will not be send or displayed, it will save into folder named template**
    b. Create macro that will generate email from a template, that has been written in cell B18 – B21
        i. Email will look like source template
        ii. Send to Name will change in body of email and in subject according to source
        iii. Where to, when will change according to source
        iv. Send to name will be red format, weekend will be bold, where to/when will be yellow
        v. **Mail will not be send or displayed, it will save into folder named template**
    c. Create macro that will generate email from an OFT template
        i. Email will look like source template
        ii. Send to Name will change in body of email and in subject according to source
        iii. Where to, when will change according to source
        iv. Send to name will be red format, weekend will be bold, where to/when will be yellow
        v. **Send the email to yourself, as otherwise you'll lose formatting when saved.**

5. *Additional knowledge*

More knowledge on email generation –

https://docs.microsoft.com/en-us/office/vba/excel/concepts/working-with-other-applications/sending-email-to-a-list-of-recipients-using-excel-and-outlook

More knowledge on email generation - https://www.emailarchitect.net/easendmail/kb/vba_excel.aspx?cat=1

HTML formatting - https://www.w3schools.com/html/html_formatting.asp