

Automated Shift-Wash Emails for Amazon Operations

Context and Objective

The operations team was manually creating shift-wash emails at the end of each shift, which included copying and pasting key metrics into Outlook and attaching screenshots. This process took approximately 10–15 minutes per shift, creating inefficiencies and delaying communication of critical information.

The objective of this project was to streamline and automate the process using VBA macros to:

1. Dynamically generate email subjects based on key metrics.
2. Embed a snapshot of the operational data directly into the email body.
3. Reduce the time required to create and send these emails.

Implementation and Code

1. Sub CreateDaysEmail and Sub CreateNightsEmail

These macros handle the creation of shift-specific emails for "Days" and "Nights" shifts. They use a shared function, `CreateDynamicSubjectEmail`, to dynamically generate the email subject and body.

Code

```
' Macro for "Days" button
Sub CreateDaysEmail()
    CreateDynamicSubjectEmail "Days"
End Sub

' Macro for "Nights" button
Sub CreateNightsEmail()
    CreateDynamicSubjectEmail "Nights"
End Sub

' Common function to create dynamic subject email with embedded range as
picture in the body
Sub CreateDynamicSubjectEmail(shiftType As String)
    ' Declare Outlook variables
    Dim OutlookApp As Object
    Dim OutlookMail As Object

    ' Create Outlook objects
    Set OutlookApp = CreateObject("Outlook.Application")
    Set OutlookMail = OutlookApp.CreateItem(0)

    ' Set email subject with the common part and the current date
    Dim commonSubject As String
    commonSubject = "LGA9 Inbound Shift Wash " & Format(Date, "mm/dd/yyyy") &
" "
```

```

' Capture the range as a picture
Dim rng As Range
Set rng = Range("A1:N66") ' Adjust the range as needed
rng.CopyPicture Appearance:=xlScreen, Format:=xlBitmap

' Adjust subject based on user's selection
If shiftType = "Nights" Then
    OutlookMail.Subject = commonSubject & "Nights [" & _
        Format(Range("C6").Value, "#,##0") & " TI, " &
-
-
-
        Format(Range("E5").Value, "0.00") & " TPH, " &
-
-
-
        Format(Range("F37").Value, "0.00") & " ETI, " &
-
-
-
        Format(Range("I66").Value, "0.0%") & " to OP]"
ElseIf shiftType = "Days" Then
    OutlookMail.Subject = commonSubject & "Days [" & _
        Format(Range("C6").Value, "#,##0") & " TI, " &
-
-
-
        Format(Range("E5").Value, "0.00") & " TPH, " &
-
-
-
        Format(Range("F37").Value, "0.00") & " ETI, " &
-
-
-
        Format(Range("I66").Value, "0.0%") & " to OP]"
Else
    ' Handle other cases if needed
    MsgBox "Invalid selection. Please choose 'Nights' or 'Days'."
    Exit Sub
End If

' Add any other properties or recipients as needed
OutlookMail.To = "lga9-ib-eos@example.com"

' Paste the captured range picture into the email body using the
RangetoHTML function
OutlookMail.HTMLBody = "<html><body>" & _
    "<p>Your email content goes here.</p>" & _
    RangetoHTML(rng) & _
    "</body></html>"

' Display the email for user review
OutlookMail.Display

' Release Outlook objects from memory
Set OutlookMail = Nothing
Set OutlookApp = Nothing
End Sub

Function RangetoHTML(rng As Range)
' Function to convert a range to HTML
' Changed by Ron de Bruin 28-Oct-2006
' Working in Office 2000-2013
    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 paste 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 an 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

```

2. Efficiency Gains

- **Time Saved:** Reduced the email creation process from 10–15 minutes to less than 1 minute per shift.
- **Consistency:** Standardized the email format, ensuring uniform communication across shifts.

- **Ease of Use:** With simple buttons for "Days" and "Nights," operations managers required minimal training to adopt the solution.
-

Conclusion

This project highlights how automation can simplify repetitive processes, saving time and ensuring accuracy in operational communications. The macros developed for shift-wash emails not only eliminated manual effort but also enhanced the quality and timeliness of information sharing within the team.