

Generic Outlook Rules via VBA

```
#If VBA7 Then
'Code is running VBA7 (2010 or later).

    #If Win64 Then
    'Code is running in 64-bit version of Microsoft Office.
    Private Declare PtrSafe Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
    #Else
    'Code is running in 32-bit version of Microsoft Office.
    Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
    #End If

#Else
'Code is running VBA6 (2007 or earlier).
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

#End If

Option Explicit

Public WithEvents objInboxItems As Outlook.Items
Private Const olOriginator As Long = 0, olTo As Long = 1, olCC As Long = 2, olBCC As Long = 3
Public AttachmentFileNamesDic As Object, olFoldersDic As Object, ExtraFiltersDic As Object
Private IgnoreFoldersArr() As String
Public objBodyStr As String, objSubjectStr As String, objSenderFullAddressStr As String, objRecipientToStr As String, objRecipientCCStr As String, objRecipientBCCStr As String
Public strFolderPath As String
Private Sub Application_Startup()
    Set objInboxItems = Session.GetDefaultFolder(olFolderInbox).Items
End Sub
Public Function GetExtraFiltersDic() As Object
    Set GetExtraFiltersDic = CreateObject("scripting.dictionary")
    GetExtraFiltersDic.CompareMode = 1

    'Personal
    GetExtraFiltersDic.Add Key:="domain.com", Item:="PERSONAL"
End Function
Private Sub objInboxItems_ItemAdd(ByVal Item As Object)
    If Item.Class = olMail Then
        Dim k, x
        Dim i As Integer, c As Integer
        Dim objMail As Outlook.MailItem
        Set objMail = Item

        Call SetMailItemVars(objMail)

        strFolderPath = "\\nawhtws002\Bradley.Vanderleeuw$\MyDocuments\Scanned Docs\"

        Set AttachmentFileNamesDic = CreateObject("scripting.dictionary")
        AttachmentFileNamesDic.CompareMode = 1
        Set AttachmentFileNamesDic = GetAttachmentFileNamesDic(objMail)

        Set ExtraFiltersDic = CreateObject("scripting.dictionary")
        ExtraFiltersDic.CompareMode = 1
        Set ExtraFiltersDic = GetExtraFiltersDic()

        Set olFoldersDic = CreateObject("scripting.dictionary")
        olFoldersDic.CompareMode = 1
        Set olFoldersDic = GetolFoldersDic()

        'If Any part of Address (Prefix, Suffix, FullAddress) are in ExtraFiltersDic then MoveOLItem
        Dim AddressSplitArr() As String, AllAddressStr As String
        AllAddressStr = objRecipientToStr & objRecipientCCStr & objRecipientBCCStr & objSenderFullAddressStr
        Erase AddressSplitArr
        AddressSplitArr = Split(AllAddressStr, ";")
        i = 0
        c = 0
        For i = LBound(AddressSplitArr) + 1 To UBound(AddressSplitArr) Step 1
```

```

c = i - 1
If ExtraFiltersDic.Exists(AddressSplitArr(c)) = True Then
    Call MoveOLItem(objMail, olFoldersDic(ExtraFiltersDic(AddressSplitArr(c))))
    Debug.Print "Moved via ExtraFiltersDic - Address Function"
    Exit Sub
ElseIf ExtraFiltersDic.Exists(GetEMailPrefixStr(AddressSplitArr(c)) = True Then
    Call MoveOLItem(objMail, olFoldersDic(ExtraFiltersDic(GetEMailPrefixStr(AddressSplitArr(c))))
    Debug.Print "Moved via ExtraFiltersDic - Address Function"
    Exit Sub
ElseIf ExtraFiltersDic.Exists(GetDomainStr(AddressSplitArr(c)) = True Then
    Call MoveOLItem(objMail, olFoldersDic(ExtraFiltersDic(GetDomainStr(AddressSplitArr(c))))
    Debug.Print "Moved via ExtraFiltersDic - Address Function"
    Exit Sub
End If
Next i

Dim SubjectSplitArr() As String
Erase SubjectSplitArr
SubjectSplitArr = Split(objSubjectStr, " ")
'If Subject is One Word and Matches ExtraFiltersDic then MoveOLItem
If UBound(SubjectSplitArr) = 0 Then
    If ExtraFiltersDic.Exists(SubjectSplitArr(0)) = True Then
        Call MoveOLItem(objMail, olFoldersDic(ExtraFiltersDic(SubjectSplitArr(0))))
        Debug.Print "Moved via ExtraFiltersDic - Subject Function"
        Exit Sub
    End If
End If
i = 0
c = 0
'If up to Three Words match Key in ExtraFiltersDic then MoveOLItem
For i = LBound(SubjectSplitArr) + 1 To UBound(SubjectSplitArr) Step 1
    c = i - 1
    If ExtraFiltersDic.Exists(SubjectSplitArr(c)) = True Then
        Call MoveOLItem(objMail, olFoldersDic(ExtraFiltersDic(SubjectSplitArr(c))))
        Debug.Print "Moved via ExtraFiltersDic - Subject Function"
        Exit Sub
    End If
    If UBound(SubjectSplitArr) > 1 And c + 1 <= UBound(SubjectSplitArr) Then
        If ExtraFiltersDic.Exists(SubjectSplitArr(c) & " " & SubjectSplitArr(c + 1)) = True Then
            Call MoveOLItem(objMail, olFoldersDic(ExtraFiltersDic(SubjectSplitArr(c) & " " & SubjectSplitArr(c + 1))))
            Debug.Print "Moved via ExtraFiltersDic - Subject Function"
            Exit Sub
        End If
    End If
    If UBound(SubjectSplitArr) >= 2 And c + 2 <= UBound(SubjectSplitArr) Then
        If ExtraFiltersDic.Exists(SubjectSplitArr(c) & " " & SubjectSplitArr(c + 1) & " " & SubjectSplitArr(c + 2))
= True Then
            Call MoveOLItem(objMail, olFoldersDic(ExtraFiltersDic(SubjectSplitArr(c) & " " & SubjectSplitArr(c + 1) &
" " & SubjectSplitArr(c + 2))))
            Debug.Print "Moved via ExtraFiltersDic - Subject Function"
            Exit Sub
        End If
    End If
End If
Next

For Each k In olFoldersDic.Keys
    If StrRegexStrickCheck(objSubjectStr, CStr(Trim(k))) = True Then 'If FolderName is in Subject move to Folder
        Call MoveOLItem(objMail, olFoldersDic(k))
        Debug.Print "Moved via olFoldersDic"
        Exit Sub
    End If
For Each x In AttachmentFileNamesDic.Keys ' If FolderName is in Attachment move to folder
    If StrRegexStrickCheck(CStr(Trim(x)), CStr(Trim(k))) = True Then
        Call MoveOLItem(objMail, olFoldersDic(k))
        Debug.Print "Moved via olFoldersDic"
        Exit Sub
    End If
Next x
Next k

Debug.Print "Item is olMail.Class"

```

```

End If
Debug.Print "No Rule Caught for objSubjectStr = " & objSubjectStr
End Sub
Public Sub MoveOLItem(objMail As Outlook.MailItem, FolerPathStr As String)
If FolerPathStr = "DOWNLOADATTACHMENTS" Then
Call DownloadObjAttachments(objMail, strFolderPath)
Debug.Print "Download Attachments"
ElseIf FolerPathStr = "JUNK" Then
objMail.UnRead = False
objMail.Delete
Debug.Print "JUNK"
ElseIf FolerPathStr <> "" Then
Dim olFolder As Outlook.Folder
Set olFolder = GetFolder(FolerPathStr)
Sleep 5000
objMail.Move olFolder
Debug.Print "Moved Item to " & FolerPathStr
ElseIf FolerPathStr = "" Then
Debug.Print "FolderPathStr = ''"
End If
End Sub
Public Function StrRegexStrickCheck(ByVal inputString As String, testName As String)
StrRegexStrickCheck = False
With CreateObject("vbscript.regexp")
.Global = True
.MultiLine = True
.IgnoreCase = True
.Pattern = "\b" & testName & "\b"
If .Test(inputString) Then StrRegexStrickCheck = True
End With
End Function
Public Sub SetMailItemVars(objMail As Outlook.MailItem)
objBodyStr = objMail.Body
'Debug.Print "objBodyStr = " & objBodyStr
objSubjectStr = objMail.Subject
'Debug.Print "objSubjectStr = " & objSubjectStr

Dim Arr As Variant: Arr = EmailAddressInfo(objMail)
objRecipientToStr = Arr(olTo)
'Debug.Print "objRecipientToStr = " & objRecipientToStr
objRecipientCCStr = Arr(olCC)
'Debug.Print "objRecipientCCStr = " & objRecipientCCStr
objRecipientBCCStr = Arr(olBCC)
'Debug.Print "objRecipientBCCStr = " & objRecipientBCCStr
objSenderFullAddressStr = Arr(olOriginator)
'Debug.Print "objSenderFullAddressStr = " & objSenderFullAddressStr

End Sub
Public Sub DownloadObjAttachments(objMail As Outlook.MailItem, strFolderPath As String)
Dim objAttachment As Attachment
For Each objAttachment In objMail.Attachments
objAttachment.SaveAsFile strFolderPath & objAttachment.FileName
Next
objMail.UnRead = False
objMail.Delete
End Sub
Public Function GetAttachmentFileNamesDic(objMail As Outlook.MailItem) As Object
Set GetAttachmentFileNamesDic = CreateObject("scripting.dictionary")
GetAttachmentFileNamesDic.CompareMode = 1
Dim tmp As String, objAttachment As Attachment
On Error Resume Next
For Each objAttachment In objMail.Attachments
tmp = Trim(objAttachment.FileName)
If Not GetAttachmentFileNamesDic.Exists(tmp) Then
GetAttachmentFileNamesDic.Add Key:=tmp, Item:="Attachment"
End If
Next objAttachment
End Function
Public Function GetDomainStr(objFullAddressStr As String) As String
Dim SplitArr() As String
SplitArr = Split(objFullAddressStr, "@")

```

```

    GetDomainStr = SplitArr(1)
End Function
Public Function GetEmailPrefixStr(objFullAddressStr As String) As String
    Dim SplitArr() As String
    SplitArr = Split(objFullAddressStr, "@")
    GetEmailPrefixStr = SplitArr(0)
End Function
Public Function GetolFoldersDic() As Object

    Set GetolFoldersDic = CreateObject("scripting.dictionary")
    GetolFoldersDic.CompareMode = 1
    Dim olApp As Outlook.Application
    Set olApp = CreateObject("outlook.application")

    Dim olSession As Outlook.NameSpace
    Set olSession = olApp.GetNamespace("MAPI")

    Dim olStartFolder As Outlook.MAPIFolder
    Set olStartFolder = olSession.GetDefaultFolder(olFolderInbox).Parent

    IgnoreFoldersArr = Split("Social Activity Notifications,ExternalContacts,Conversation Action Settings,
    PersonMetadata,Journal,Quick Step Settings,RSS Subscriptions,Archive,Notes,Sync Issues,Yammer Root,Files,Tasks,
    Contacts,Calendar,Conversation History,Drafts,Junk Email,Sent Items,Outbox,Inbox,Deleted Items,Team Chat,
    Birthdays,United States holidays,Recipient Cache,Skype for Business Contacts,PeopleCentricConversation Buddies,
    Organizational Contacts,GAL Contacts,Companies,Feeds,Inbound,Outbound,Local Failures,Server Failures,
    Conflicts",",")

    Dim olNewFolder As Outlook.MAPIFolder, i As Long, olFolderStr As String, olFolderPath As String
    For Each olNewFolder In olStartFolder.Folders
        For IsStrInArr(IgnoreFoldersArr, olNewFolder.Name) = False Then
            If olNewFolder.Folders.Count > 0 Then
                For i = olNewFolder.Folders.Count To 1 Step -1
                    olFolderStr = Trim(olNewFolder.Folders(i))
                    olFolderPath = olNewFolder.Folders(i).FolderPath
                    If Not GetolFoldersDic.Exists(olFolderStr) Then
                        GetolFoldersDic.Add Key:=olFolderStr, Item:=olFolderPath
                    End If
                Next
            Else
                olFolderStr = Trim(olNewFolder.Name)
                olFolderPath = olNewFolder.FolderPath
                If Not GetolFoldersDic.Exists(olFolderStr) Then
                    GetolFoldersDic.Add Key:=olFolderStr, Item:=olFolderPath
                End If
            End If
        Next
        GetolFoldersDic.Add Key:="DOWNLOADATTACHMENTS", Item:="DOWNLOADATTACHMENTS"
        GetolFoldersDic.Add Key:="JUNK", Item:="JUNK"
    End Function
Public Function IsStrInArr(Arr As Variant, StrToCheck As String) As Boolean
    IsStrInArr = UBound(Filter(Arr, StrToCheck)) > -1
End Function

Private Function EmailAddressInfo(olItem As MailItem) As Variant
    If olItem.Class <> olMail Then Exit Function

On Error GoTo ExitFunction

    Dim olRecipient As Outlook.Recipient
    Dim olEU As Outlook.ExchangeUser
    Dim olEDL As Outlook.ExchangeDistributionList
    Dim ToAddress, CCAddress, BCCAddress, Originator, email As String

    With olItem
        Select Case UCase(.SenderEmailType)
            Case "SMTP": Originator = .SenderEmailAddress
            Case Else
                Set olEU = .Sender.GetExchangeUser
                If Not olEU Is Nothing Then Originator = olEU.PrimarySmtpAddress
        End Select

```

```

End With

For Each olRecipient In olItem.Recipients
    With olRecipient
        Select Case .AddressEntry.AddressEntryUserType
            Case olSmtptAddressEntry 'OlAddressEntryUserType.
                email = .Address
            Case olExchangeDistributionListAddressEntry, olOutlookDistributionListAddressEntry
                Set olEDL = .AddressEntry.GetExchangeDistributionList
                email = IIf(Not olEDL Is Nothing, olEDL.PrimarySmtptAddress, "")
            Case Else
                Set olEU = .AddressEntry.GetExchangeUser
                email = IIf(Not olEU Is Nothing, olEU.PrimarySmtptAddress, "")
        End Select
        If email <> "" Then
            Select Case .Type
                Case olTo: ToAddress = ToAddress & email & ";"
                Case olCC: CCAddress = CCAddress & email & ";"
                Case olBCC: BCCAddress = BCCAddress & email & ";"
            End Select
        End If
    End With
Next
EmailAddressInfo = Array(Originator, ToAddress, CCAddress, BCCAddress)
ExitFunction:
End Function

Function GetFolder(ByVal FolderPath As String) As Outlook.Folder
    Dim TestFolder As Outlook.Folder
    Dim FoldersArray As Variant
    Dim i As Integer

    On Error GoTo GetFolder_Error
    If Left(FolderPath, 2) = "\\\" Then
        FolderPath = Right(FolderPath, Len(FolderPath) - 2)
    End If

    'Convert folderpath to array
    FoldersArray = Split(FolderPath, "\")
    Set TestFolder = Application.Session.Folders.Item(FoldersArray(0))
    If Not TestFolder Is Nothing Then
        For i = 1 To UBound(FoldersArray, 1)
            Dim SubFolders As Outlook.Folders
            Set SubFolders = TestFolder.Folders
            Set TestFolder = SubFolders.Item(FoldersArray(i))
            If TestFolder Is Nothing Then
                Set GetFolder = Nothing
            End If
        Next
    End If

    'Return the TestFolder
    Set GetFolder = TestFolder
    Exit Function

GetFolder_Error:
    Set GetFolder = Nothing
    Exit Function
End Function

```