

## Code:Import Emails into Access

**‘Note: This code is part of a web post on our blog [AccessExperts.net/blog](http://AccessExperts.net/blog). Visit the article first if you stumbled upon this file. You will also need to add a reference to Outlook in Access.**

```
Private Sub Form_Load()
```

```
‘The load procedure will read all emails in the Import folder of Outlook and then present the list to the user for selection
```

```
Dim objApp As Outlook.Application
```

```
Dim objNS As Outlook.Namespace
```

```
Dim colFolders As Outlook.Folders
```

```
Dim objFolder As Outlook.MAPIFolder
```

```
Dim objParentFolder As Outlook.MAPIFolder
```

```
Dim objMailItem As Outlook.MailItem
```

```
Dim strSQL As String
```

```
Dim strBody As String
```

```
Dim strSubject As String
```

```
Const strDoubleQuote As String = """"
```

```
Dim strFolderPath As String
```

```
Dim i As Byte
```

```
On Error GoTo Form_Load_Error
```

‘I use EmptyTable to run SQL code “Delete from tblImportEmails”. Search for EmptyTable below for the code  
EmptyTable "tblImportEmails"

```
Set objApp = New Outlook.Application
```

```
Set objNS = objApp.GetNamespace("MAPI")
```

```
For i = 1 To objNS.Folders.Count
```

```
Set objParentFolder = objNS.Folders(i)
```

```
‘I use OutlookFolderNames to locate the folder in the user’s Outlook
```

```
Set objFolder = OutlookFolderNames(objParentFolder, "Import")
```

```
If Not objFolder Is Nothing Then
```

```
‘Once the folder is located exit For
```

```
Exit For
```

```
End If
```

‘what happens when there is no folder in outlook? code stops and here you need to check if user has Import folder in their Outlook

```
Next i
```

```
If objFolder Is Nothing Then
```

```
MsgBox "Please make sure you have two folders in your Outlook:" & vbCrLf & _
```

```
"Import" & vbCrLf & _
```

```
"Imported.", vbInformation, "Missing Folders in Outlook"
```

```
DoCmd.Close acForm, Me.Name
```

```
Exit Sub
```

```
End If
```

```
If Not objFolder Is Nothing Then
```

```
For Each objMailItem In objFolder.Items
```

```
With objMailItem
```

**‘I build the SQL statement to save email details in temp folder, notice the user of replace to take care of single quotes and other issues.**

```

strSQL = "From: " & .SenderName & " (" & Replace(.SenderEmailAddress, "", "") & ")" & vbCrLf
strSQL = strSQL & "To: " & Replace(.To, "", "") & vbCrLf
strSQL = strSQL & "CC: " & Replace(.CC, "", "") & vbCrLf
strSubject = Replace(.Subject, "", "")
'strSubject = Replace(strSubject, Chr(34), Chr(34) & Chr(34))

```

```

strSQL = strSQL & "Subject: " & strSubject & vbCrLf
strSQL = strSQL & "Date Emailed: " & .SentOn & vbCrLf

```

```

strBody = Replace(.Body, "", "")
strBody = Replace(strBody, Chr(34), Chr(34) & Chr(34))
strSQL = strSQL & strBody

```

```

strSQL = "Insert Into tblImportEmails(EmailFrom, EmailTo, EmailCC, EmailSubject, EmailBody, EmailDate, Message,
OutlookID) " & _
    "Values(" & .SenderEmailAddress & ", " & Replace(.To, "", "") & ", " & Replace(.CC, "", "") & ", " & strSubject & _
    ", " & strBody & ",#" & _
    ".SentOn & ",#" & strSQL & ", " & .EntryID & ")"
CurrentDb.Execute strSQL
End With
Next objMailItem
End If

```

'This next step will populate the list on the form to display emails ready for Import, notice how we use OutlookID to identify each message.

```

Me.lstEmails.RowSource = "Select ImportEmailID, EmailFrom, EmailSubject, EmailBody, OutlookID From tblImportEmails"
Set objFolder = Nothing
Set objNS = Nothing
Set objApp = Nothing

```

```

ExitProcedure:
On Error GoTo 0
Exit Sub

```

Form\_Load\_Error:

```

MsgBox "Error " & Err.Number & " (" & Err.Description & ") in procedure Form_Load of VBA Document
Form_frmImportEmails"
End Sub

```

```

Private Sub cmdImport_Click()

```

**'This procedure is attached to a command button on the import form in Access. It's clicked after user has selected multiple items to be imported.**

```

Dim objApp As Outlook.Application
Dim objNS As Outlook.Namespace
Dim colFolders As Outlook.Folders
Dim objFolder As Outlook.MAPIFolder
Dim objParentFolder As Outlook.MAPIFolder
Dim objDestFolder As Outlook.MAPIFolder
Dim objMailItem As Outlook.MailItem
Dim strSQL As String
Dim strFolderPath As String
Dim i As Byte

```

```

Dim varitem As Variant
Dim rs As DAO.Recordset
Dim rsJobNote As ADODB.Recordset
Dim rsJobNoteID As ADODB.Recordset
Dim strBody As String

```

```

On Error GoTo cmdImport_Click_Error

```

```

If Me.lstEmails.ItemsSelected.Count = 0 Then
    MsgBox "You haven't selected any emails to import, please make your selection and try again.", vbInformation, "No Selection Made"
    Exit Sub
End If

```

#### **'Add email body to database table tblJobNotes**

```

For Each varitem In Me.lstEmails.ItemsSelected
    strSQL = "Select Message, JobNoteID from tblImportEmails Where ImportEmailID = " & Me.lstEmails.Column(0, varitem)
    Set rs = CurrentDb.OpenRecordset(strSQL)
    strBody = Replace(rs!Message, "", "")
    strBody = Replace(strBody, Chr(34), Chr(34) & Chr(34))
    strSQL = "Insert Into tblJobNotes(JobID, JobNote, CreatedBy) " & _
        "Values(" & Me.OpenArgs & "," & strBody & "," & GetNetworkName() & ");SELECT SCOPE_IDENTITY() as NewJobLetterID"
    If con.State = adStateClosed Then
        OpenMyConnection
    End If
    Set rsJobNote = con.Execute(strSQL)
    Set rsJobNoteID = rsJobNote.NextRecordset
    With rs
        .Edit
        !JobNoteID = rsJobNoteID!NewJobLetterID
        .Update
    End With
Next varitem

```

```

Set objApp = New Outlook.Application
Set objNS = objApp.GetNamespace("MAPI")
For i = 1 To objNS.Folders.Count
    Set objParentFolder = objNS.Folders(i)
    'Debug.Print objParentFolder.Name
    Set objFolder = OutlookFolderNames(objParentFolder, "JobTrackerIn")
    If Not objFolder Is Nothing Then
        Exit For
    End If
    'what happens when there is no folder in outlook?
Next i

```

```

For i = 1 To objNS.Folders.Count
    Set objParentFolder = objNS.Folders(i)
    'Debug.Print objParentFolder.Name
    Set objDestFolder = OutlookFolderNames(objParentFolder, "JobTrackerProcessed")
    If Not objDestFolder Is Nothing Then
        Exit For
    End If
Next i

```

```

End If
'what happens when there is no folder in outlook?
Next i

' Set objDestFolder = objNS.Folders(strParentFolder).Folders("JobTrackerProcessed")
If Not objFolder Is Nothing Then
StartAgain:
  For i = 1 To objFolder.Items.Count
    Set objMailItem = objFolder.Items(i)
    With objMailItem
      'was the item selected?
      For Each varitem In Me.lstEmails.ItemsSelected
        If .EntryID = Me.lstEmails.Column(4, varitem) Then
          'If email has attachments then save those as well
          If .Attachments.Count > 0 Then
            'If there are any attachments then use a separate procedure to deal with them. See below
            CopyAttachments objMailItem
          End If
          .Move objDestFolder
          GoTo StartAgain
        End For
      End If
    Next varitem
  End With
Next i
End If

Set objFolder = Nothing
Set objParentFolder = Nothing
Set objNS = Nothing
Set objApp = Nothing

Set rs = Nothing
DoCmd.Close acForm, Me.Name

ExitProcedure:
On Error GoTo 0
Exit Sub

cmdImport_Click_Error:

MsgBox "Error " & Err.Number & " (" & Err.Description & ") in procedure cmdImport_Click of VBA Document
Form_frmImportEmails"
Resume ExitProcedure
Resume
End Sub

Private Sub CopyAttachments(objMailItem As Outlook.MailItem)
Dim myAttachments As Outlook.Attachments
Dim i As Byte
Dim Job As clsJob
Dim strJobDirectory As String

```

```
Dim rs As ADODB.Recordset
Dim rsNotes As DAO.Recordset
Dim strSQL As String
Dim rsDoc As ADODB.Recordset
Dim lngDocID As Long
```

'Job is what the database is about, the class below will return the directory on the network where to store docs

```
Set Job = New clsJob
With Job
    .JobID = Me.OpenArgs
    strJobDirectory = .JobDirectory
End With
```

```
Set rsNotes = CurrentDb.OpenRecordset("Select JobNoteID from tblImportEmails Where OutlookID = " &
objMailItem.EntryID & """)
```

```
'Update job notes table to set HasAttachments = True
strSQL = "Update tblJobNotes Set HasAttachments = 1 Where JobNoteID = " & rsNotes!JobNoteID
ExecuteMyCommand strSQL
```

```
Set myAttachments = objMailItem.Attachments
With myAttachments
    For i = 1 To .Count
```

**'I notate in the message if it came with attachments, which the user can bring up using Doc Center**

```
    strSQL = "Insert Into tblJobLetters(LetterTypeID, JobID, CreatedBy, JobNoteID, Note) Values(" &
conDocTypeEmailAttachment & ", " & _
    Me.OpenArgs & ", " & GetNetworkName() & ", " & rsNotes!JobNoteID & ", " & myAttachments(i).DisplayName & ");" &
```

```
    "Select SCOPE_IDENTITY() as NewDocID"
```

```
    If con.State = adStateClosed Then
        OpenMyConnection
```

```
    End If
    Set rsDoc = con.Execute(strSQL)
    lngDocID = rsDoc.NextRecordset!NewDocID
    Set rsDoc = Nothing
```

```
    strSQL = "Update tblJobLetters Set DocPath = " & strJobDirectory & lngDocID & myAttachments(i).DisplayName & ""
Where JobLetterID = " & lngDocID
    ExecuteMyCommand strSQL
    'Save attachment to job directory
    myAttachments(i).SaveAsFile strJobDirectory & lngDocID & myAttachments(i).DisplayName
    Next i
End With
```

```
Set Job = Nothing
Set myAttachments = Nothing
```

```
End Sub
```

```
Private Function OutlookFolderNames(objFolder As Outlook.MAPIFolder, strFolderName As String) As Object
```

```
*****
```

```
On Error GoTo ErrorHandler
Dim objOneSubFolder As Outlook.MAPIFolder
Dim oFolder As Outlook.Folder
```

```

If Not objFolder Is Nothing Then
  If LCase(strFolderName) = LCase(objFolder.Name) Then
    Set OutlookFolderNames = objFolder
  Else
    ' Check if folders collection is not empty
    If objFolder.Folders.Count > 0 And _
      Not objFolder.Folders Is Nothing Then
      For Each oFolder In objFolder.Folders
        Set objOneSubFolder = oFolder
        ' only check mail item folder
        If objOneSubFolder.DefaultItem Type _
          = olMailItem Then
          If LCase(strFolderName) = _
            LCase(objOneSubFolder.Name) Then
            Set OutlookFolderNames = _
              objOneSubFolder
            Exit For
          Else
            If objOneSubFolder.Folders.Count _
              > 0 Then
              Set OutlookFolderNames = _
                OutlookFolderNames _
                  (objOneSubFolder, strFolderName)
              If Not (OutlookFolderNames Is Nothing) Then
                MsgBox "It worked!"
                Exit Function
              End If
            End If
          End If
        End If
      Next
    End If
  End If
End If

```

```
Exit Function
```

```
ErrorHandler:
```

```
Set OutlookFolderNames = Nothing
```

```
End Function
```

```
Public Sub EmptyTable(strTable As String)
```

```

'-----
' Procedure : EmptyTable
' DateTime  : 7/20/2006 14:20
' Author   : Juan Soto/ AccessExperts.net/blog
' Purpose  : Empty the table in the database represented by strTable. Usually used for temporary tables.
'-----

```

```
'REVISED
```

```
'Date By Comment
```

```
'
```

```
Dim strSQL As String
```

```
On Error GoTo EmptyTable_Error
```

```
strSQL = "Delete * From " & strTable
CurrentDb.Execute strSQL, dbSeeChanges
```

```
On Error GoTo 0
Exit Sub
```

```
EmptyTable_Error:
```

```
End Sub
```

```
'Place this function in a global module
```

```
Public Function OpenMyConnection() As Boolean
```

```
'Note: Use Dim con As ADODB.Connection in a public module to have a global connection object in your project.
```

```
If con.State = adStateOpen Then
    con.Close
```

```
End If
```

```
con.ConnectionString = conConnection & "User ID =" & ReadGV("UserID", strText) & ";Password=" & ReadGV("Password",
strText)
```

```
con.Open
```

```
If Not con.State = adStateOpen Then
```

```
    OpenMyConnection = False
```

```
Else
```

```
    OpenMyConnection = True
```

```
End If
```

```
End Function
```

```
'Place this function in a global module
```

```
Public Function GetNetworkName() As String
```

```
Dim ret As Long
```

```
Dim cbusername As Long
```

```
Dim UserName As String
```

```
Dim computername As String
```

```
Dim logondomain As String
```

```
Dim langroup As String
```

```
On Error GoTo ErrorHandler
```

```
' Clear all of the display values.
```

```
computername = "": langroup = "": UserName = "": logondomain = ""
```

```
' Windows 95 or NT - call WNetGetUser to get the name of the user.
```

```
UserName = Space(256)
```

```
cbusername = Len(UserName)
```

```
ret = WNetGetUser(ByVal 0&, UserName, cbusername)
```

```
"Debug.Print cbusername
```

```
If ret = 0 Then
```

```
    ' Success - strip off the null.
```

```
    UserName = Left(UserName, InStr(UserName, Chr(0)) - 1)
```

```
Else
```

```
    GoTo ExitProcedure
```

```
End If
```

```
If UserName = "juan" Or UserName = "Administrator" Then
    GetNetworkName = "dbo"
Else
    GetNetworkName = UserName
End If
```

ExitProcedure:

```
Exit Function
```

ErrorHandler:

```
MsgBox Err.Description
```

```
MsgBox "Could not validate " & UserName & " user name. You will not be able to use this system.", vbCritical, "Access Denied"
```

```
Application.Quit
```

```
Resume ExitProcedure
```

```
Resume Next
```

End Function

---

Published by [Google Drive](#) – [Report Abuse](#) – Updated automatically every 5 minutes

---