

MS Outlook: How to Move Mail Items to a Folder via One Single Click



Applies to:

This article applies to Microsoft Outlook technology and is - in my opinion - release independent. I'm personally running Outlook 2007 SP2, but other versions should be supported as well.

This article is not related to any SAP technology but as an important number of SAP SDN community members are using Microsoft products (in particular MS Outlook) on a daily base it might be useful for them.

Summary

Do you struggle as well classifying all of your incoming mails? I did, which is why I decided to create and add a dedicated button bar to my MS Outlook application that allows me to move single or multiple mail items in specific folder.

In the past I had to browse through a complex multi-level structure of Outlook folders before I was able to move mail items from my default Inbox to any 'archive' folder. This was not only time consuming but also frustrating because I couldn't always recall the name of the folder. Luckily this belongs to the past, thanks to the simple VBA macro that I wrote.

Author: Danny De Roovere

Company: SAP Belgium

Created on: 18 May 2010

Author Bio



Danny De Roovere is a SAP Certified consultant (PI, Portal, Solution Manager, TOGAF) employed at SAP Belgium-Luxembourg with a focus on Enterprise Architecture, (technical) project management, the SAP business process platform, SAP PI administration & development, SAP Enterprise Portal administration & development, SAP Application Management, ALE & EDI

Table of Contents

Introduction	3
Input file	3
VBA Code	4
Assign Macro to Button	9
Result	10
Related Content	11
Copyright	12

Introduction

Have you ever wondered how you could easily move mail items from your default Inbox to a structured set of folders? I did, and found the solution. It is so convenient, which is why I want to share the VBA code with you. The solution exists of a simple set of VBA instructions running as a Macro in your Outlook application. It takes a text file as input and creates one button for each line item in the text file.

Input file

Below you find a sample of the text file.

You can give the file any name you want and store it wherever you want. You only need to change the full path reference to the file in the VBA code as described in the next chapter.

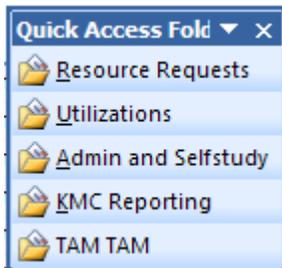
```
&Resource Requests,EMEA short term resource requests,Danny De Roovere/4. Knowledge Transfer NW/Team Coordination/Resource Requests and Soft bookings
&Utilizations,Team Utilizations,Danny De Roovere/4. Knowledge Transfer NW/Team Coordination/Consulting/Utilizations
&Admin and Selfstudy,Team Administration and Selfstudy activities,Danny De Roovere/4. Knowledge Transfer NW/Team Coordination/Consulting/Admin and Selfstudy
&KMC Reporting,Knowledge Management Champion reporting,Danny De Roovere/4. Knowledge Transfer NW/KM Champion/Reporting
TAM TAM,SAP Belux TAM TAM,Danny De Roovere/2. SAP Info/TamTam
```

It is an easy-to-maintain solution for adding/removing/updating information about the Outlook folder structure and the shortcuts (read: buttons) that you want to assign. If you want to remove buttons, you delete lines. If you desire to add buttons you just add additional line items etc.

Each line item represents one button

The line items consists of three parts, each part is separated by a comma. The 1st part contains the label of the button where the ampersand is used for the shortcut assignment (ALT + key). The second part contains the description of the folder and the third part contains the full path to the Outlook mail folder. The full path can contain a personal mail folder, the default mail folder or a public mail folder. If the mail folder does not exist in the existing tree of Outlook folders, the Macro will prompt the user to create it.

The above sample is used to create a button bar with 5 buttons:



VBA Code

Open Outlook and “cut-and-paste” the following code via Tools > Macro > Create.

The code doesn't need any further explanation as the comments are included.

```

Sub ToggleQuickAccessFolderToolBar()
    'Author: Danny De Roovere - SAP Belux 2010
    'Macro Purpose: Function checks whether "Quick Access Folder" command bar exists
    '                If it exists & the command bar is visible then the command bar is
made invisible (and vice versa)
    '                If it does not exists the command bar is created. In the latter case
a file containing the button
    '                information must exist.

    Dim cb As CommandBar
    Dim sPath As String
    Set myExplorer = Application.ActiveExplorer

    For Each cb In myExplorer.CommandBars

        If cb.Name = "Quick Access Folder" Then
            If cb.Visible = True Then
                cb.Visible = False
                GoTo ExitToggle
            ElseIf cb.Visible = False Then
                cb.Visible = True
                GoTo ExitToggle
            End If
        End If
    Next

    AddCommandBar:
    sPath = "C:\users\Development\VBA\Move Mail Items\OLFolders.txt"
    If FileOrDirExists(sPath) Then
        Set CommandBar = myExplorer.CommandBars.Add(Name:="Quick Access Folder", _
            Position:=msoBarFloating, Temporary:=True)
        ' Position:=msoBarTop, Temporary:=True)
        CommandBar.Visible = True
        Call ReadQuickAccessFolderTextFile(sPath)
    Else
        MsgBox "The CSV file containing the Quick Access Folders info '" & sPath & "'
does not exist."
    End If

```



```
ExitToggle:
```

```
End Sub
```

```
Function FileOrDirExists(PathName As String) As Boolean
```

```
    'Macro Purpose: Function returns TRUE if the specified file  
    '                  or folder exists, false if not.
```

```
    'PathName      : Supports Windows mapped drives or UNC
```

```
    'File usage    : Provide full file path and extension
```

```
    'Folder usage  : Provide full folder path
```

```
    '              : Accepts with/without trailing "\" (Windows)
```

```
    Dim iTemp As Integer
```

```
    'Ignore errors to allow for error evaluation
```

```
    On Error Resume Next
```

```
    iTemp = GetAttr(PathName)
```

```
    'Check if error exists and set response appropriately
```

```
    Select Case Err.Number
```

```
    Case Is = 0
```

```
        FileOrDirExists = True
```

```
    Case Else
```

```
        FileOrDirExists = False
```

```
    End Select
```

```
    'Resume error checking
```

```
    On Error GoTo 0
```

```
End Function
```

```
Function ReadQuickAccessFolderTextFile(PathName As String)
```

```
    'Macro Purpose: Function reads a CSV File with structure ButtonLabel, Folder  
    Description, Outlook Folder
```

```
    '                  and adds a button for each Teamco to the "Quick Access Folder"  
button bar
```

```
    Dim ButtonLabel, FolderDesc, OLFolder As String
```

```
    Dim caption As String
```

```
    Dim Folder As String
```

```
On Error Resume Next
```

```
Open PathName For Input As #1 ' Open file for input.
```

```
Do While Not EOF(1) ' Loop until end of file.
```

```
    Input #1, ButtonLabel, FolderDesc, OLFolder ' Read data into three variables.
```

```
    caption = ButtonLabel
```

```
    Folder = OLFolder
```

```
    Call AddToolBarButton(caption, "Move item to folder " & FolderDesc, Folder,
"Quick Access Folder", 721, msoButtonIconAndCaption)
```

```
Loop
```

```
Close #1 ' Close file.
```

```
End Function
```

```
Function AddToolBarButton(caption As String, _
```

```
    tooltip As String, _
```

```
    Folder As String, _
```

```
    Optional toolbarName As String = "Standard", _
```

```
    Optional FaceID As Long = 325, _
```

```
    Optional buttonStyle As MsoButtonStyle = msoButtonIconAndCaption)
```

```
'Macro Purpose: Add buttons to the command bar
```

```
Dim objBar As Office.CommandBar
```

```
Dim objButton As Office.CommandBarButton
```

```
Set objBar = ActiveExplorer.CommandBars(toolbarName)
```

```
Set objButton = objBar.Controls.Add(msoControlButton, , , , True)
```

```
With objButton
```

```
    .caption = caption
```

```
    .OnAction = "GetButtonFolderPathAndMoveItem"
```

```
    .Parameter = Folder
```

```
    .TooltipText = tooltip
```

```
    .FaceID = FaceID
```

```
    .Style = buttonStyle
```

```
    .BeginGroup = True
```

```
End With
```

```
End Function
```

```
Function MoveSelectedMessagesToFolder(strFolderPath As String)
```

```

'Macro Purpose: Function moves message to Path passed as parameter
'               If the folder does not exist, the user will be warned
'-----
Dim objApp As Outlook.Application
Dim objNS As Outlook.NameSpace
Dim objItem As Outlook.MailItem
Dim colFolders As Outlook.Folders
Dim objFolder As Outlook.MAPIFolder
Dim objInbox As Outlook.MAPIFolder
Dim arrFolders() As String
Dim msg As String
Dim I As Long
On Error Resume Next

If Application.ActiveExplorer.Selection.Count = 0 Then
'At least one message must be selected when calling this procedure
Exit Function
End If

'Parse the complete Path into individual Outlook folders
strFolderPath = Replace(strFolderPath, "/", "\")
arrFolders() = Split(strFolderPath, "\")
Set objApp = Application
Set objNS = objApp.GetNamespace("MAPI")
'determine root folder (= personal folder)
Set objFolder = objNS.Folders.Item(arrFolders(0))
'check if personal folder exists
If Not objFolder Is Nothing Then
For I = 1 To UBound(arrFolders)
Set colFolders = objFolder.Folders
Set objFolder = Nothing
'construct full path
Set objFolder = colFolders.Item(arrFolders(I))
'check whether subfolder exists
If objFolder Is Nothing Then
msg = "The folder '" & arrFolders(I) & "' in the Path '" & strFolderPath & "'
doesn't exist!" & Chr(13) & "Do you want to create the folder?"
Answer = MsgBox(msg, vbOKCancel, "INVALID FOLDER")
If Answer = vbCancel Then GoTo ExitMove ' the function ends if the user selects
the CANCEL-button
colFolders.Add (arrFolders(I))
Set objFolder = colFolders.Item(arrFolders(I))

```

```

    End If
  Next
Else
  MsgBox "The Personal folder '" & arrFolders(0) & "' doesn't exist!" & Chr(13) &
  "Please, create it."
  GoTo ExitMove
End If

```

```
' move message to folder
```

```

  For Each objItem In Application.ActiveExplorer.Selection
    If objFolder.DefaultItemType = olMailItem Then
      If objItem.Class = olMail Then
        objItem.Move objFolder
      End If
    End If
  Next

```

```
ExitMove:
```

```

  Set objItem = Nothing
  Set objInbox = Nothing
  Set objFolder = Nothing
  Set objNS = Nothing
  Set colFolders = Nothing
  Set objApp = Nothing

```

```
End Function
```

```
Private Sub GetButtonFolderPathAndMoveItem()
```

```

  'Macro Purpose: macro triggered by button that takes parameter (Folder Path) from
  command bar button

```

```

  Dim objNamespace As NameSpace
  Dim objItem As Object
  Dim strEntryID As String
  strEntryID = _
    Application.ActiveExplorer.CommandBars.ActionControl.Parameter

```

```
' for debugging      MsgBox "You selected: " & strEntryID
```

```
  Call MoveSelectedMessagesToFolder(strEntryID)
```

```
End Sub
```

Assign Macro to Button

In Outlook 2007 and Outlook 2010, the Developer tab on the Ribbon, which holds the Macros menu for opened items, is disabled by default.

To enable it for Outlook 2007:

1. Open a message or create a new one.
2. Click on the Office logo in the left top and choose Editor Options.
3. In the Popular section, enable the option: Show Developer tab in the Ribbon.
4. Press OK to close the open dialog.

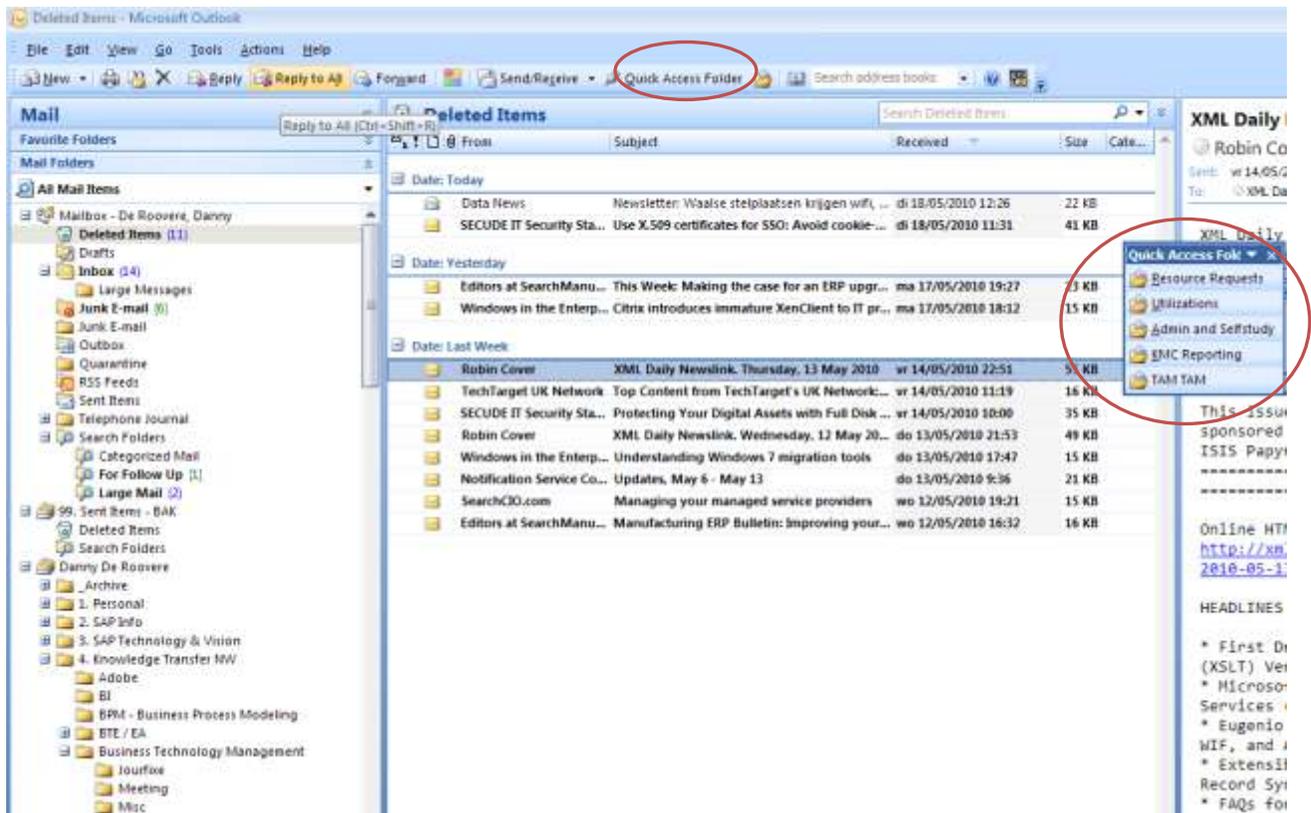
To enable it for Outlook 2010:

1. Press the Office logo button next to the Home tab and choose Options.
2. Select the section Customize Ribbon.
3. In the right pane, enable the selection field before "Developer".
4. Press OK to close the open dialog.

Assign Macro to button:

1. Tools-> Customize
2. Verify that the Commands tab is selected.
In the Categories list on the right, select Macros.
The Commands list on the right will now display all your macros
3. Drag and drop the macro that you wish to create a button for to the location on the toolbar where you want it. In our example Project1.ToggleQuickAccessFolderToolbar
4. To modify the name and icon, right click the icon on the Toolbar;
 - You can edit the name to something more meaningful.
If you want to use an ALT+key shortcut for the button, place an & for the letter that you want to use.
For instance, let's assume you call it "Quick Access Folder" and want to use the ALT+Q combination.
You'd then write "&Quick Access Folder "
 - You can change the button image via the Change Button Image
 - You can also disable the text or button or image if you don't like to see it.
5. Close the Customize dialog to leave Edit Mode of your Toolbars and use your button.

Result



Related Content

[Getting Started with Desktop Intelligence VBA macros](#)

[Basics to developing VBA Macros](#)

[How to connect to SAP R/3 and call BAPI from MS Excel VBA](#)

Copyright

© Copyright 2010 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.