

```
'-----  
' The sample scripts are not supported under any Microsoft standard support  
' program or service. The sample scripts are provided AS IS without warranty  
' of any kind. Microsoft further disclaims all implied warranties including,  
' without limitation, any implied warranties of merchantability or of fitness for  
' a particular purpose. The entire risk arising out of the use or performance of  
' the sample scripts and documentation remains with you. In no event shall  
' Microsoft, its authors, or anyone else involved in the creation, production, or  
' delivery of the scripts be liable for any damages whatsoever (including,  
' without limitation, damages for loss of business profits, business interruption,  
' loss of business information, or other pecuniary loss) arising out of the use  
' of or inability to use the sample scripts or documentation, even if Microsoft  
' has been advised of the possibility of such damages.  
'-----
```

Option Explicit

```
' *****  
' For Outlook 2010.  
' *****  
#If VBA7 Then  
    ' The window handle of Outlook.  
    Private lHwnd As LongPtr  
  
    ' /* API declarations. */  
    Private Declare PtrSafe Function FindWindow Lib "user32" Alias "FindWindowA"  
(ByVal lpClassName As String, _  
    ByVal lpWindowName As String) As LongPtr  
  
' *****  
' For the previous version of Outlook 2010.  
' *****  
#Else  
    ' The window handle of Outlook.  
    Private lHwnd As Long  
  
    ' /* API declarations. */  
    Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal  
lpClassName As String, _  
    ByVal lpWindowName As String) As Long  
#End If  
  
' The class name of Outlook window.  
Private Const olAppCLSN As String = "rctrl_renwnd32"  
' Windows desktop - the virtual folder that is the root of the namespace.  
Private Const CSIDL_DESKTOP = &H0  
' Only return file system directories. If the user selects folders that are not part  
of the file system, the OK button is grayed.  
Private Const BIF_RETURNONLYFSDIRS = &H1  
' Do not include network folders below the domain level in the dialog box's tree view  
control.  
Private Const BIF_DONTGOBELOWDOMAIN = &H2  
' The maximum length for a path is 260 characters.  
Private Const MAX_PATH = 260  
  
' #####  
' Returns the number of attachements in the selection.  
' #####
```

```

Public Function SaveAttachmentsFromSelection() As Long
    Dim objFSO           As Object           ' Computer's file system object.
    Dim objShell         As Object           ' Windows Shell application object.
    Dim objFolder        As Object           ' The selected folder object from Browse
for Folder dialog box.
    Dim objItem          As Object           ' A specific member of a Collection object
either by position or by key.
    Dim selItems         As Selection        ' A collection of Outlook item objects in
a folder.
    Dim atmt             As Attachment      ' A document or link to a document
contained in an Outlook item.
    Dim strAtmtPath      As String           ' The full saving path of the attachment.
    Dim strAtmtFullName  As String           ' The full name of an attachment.
    Dim strAtmtName(1)   As String           ' strAtmtName(0): to save the name;
strAtmtName(1): to save the file extension. They are separated by dot of an attachment
file name.
    Dim strAtmtNameTemp  As String           ' To save a temporary attachment file
name.
    Dim intDotPosition   As Integer         ' The dot position in an attachment name.
    Dim atmts            As Attachments     ' A set of Attachment objects that
represent the attachments in an Outlook item.
    Dim lCountEachItem   As Long           ' The number of attachments in each
Outlook item.
    Dim lCountAllItems   As Long           ' The number of attachments in all Outlook
items.
    Dim strFolderPath    As String           ' The selected folder path.
    Dim blnIsEnd         As Boolean         ' End all code execution.
    Dim blnIsSave        As Boolean         ' Consider if it is need to save.

    blnIsEnd = False
    blnIsSave = False
    lCountAllItems = 0

    On Error Resume Next

    Set selItems = ActiveExplorer.Selection

    If Err.Number = 0 Then

        ' Get the handle of Outlook window.
        lHwnd = FindWindow(olAppCLSN, vbNullString)

        If lHwnd <> 0 Then

            ' /* Create a Shell application object to pop-up BrowseForFolder dialog
box. */
            Set objShell = CreateObject("Shell.Application")
            Set objFSO = CreateObject("Scripting.FileSystemObject")
            Set objFolder = objShell.BrowseForFolder(lHwnd, "Select folder to save
attachments:", _
                                                    BIF_RETURNONLYFSDIRS +
BIF_DONTGOBELOWDOMAIN, CSIDL_DESKTOP)

            ' /* Failed to create the Shell application. */
            If Err.Number <> 0 Then
                MsgBox "Run-time error '" & CStr(Err.Number) & " (0x" &
CStr(Hex(Err.Number)) & ")':" & vbCrLf & _
                    Err.Description & ".", vbCritical, "Error from Attachment
Saver"

                blnIsEnd = True
                GoTo PROC_EXIT
            End If
        End If
    End If

```

```

If objFolder Is Nothing Then
    strFolderPath = ""
    blnIsEnd = True
    GoTo PROC_EXIT
Else
    strFolderPath = CGPath(objFolder.Self.Path)

    ' /* Go through each item in the selection. */
    For Each objItem In selItems
        lCountEachItem = objItem.Attachments.Count

        ' /* If the current item contains attachments. */
        If lCountEachItem > 0 Then
            Set atmts = objItem.Attachments

            ' /* Go through each attachment in the current item. */
            For Each atmt In atmts

                ' Get the full name of the current attachment.
                strAtmtFullName = atmt.FileName

                ' Find the dot position in atmtFullName.
                intDotPosition = InStrRev(strAtmtFullName, ".")

                ' Get the name.
                strAtmtName(0) = Left$(strAtmtFullName, intDotPosition -
1)

                ' Get the file extension.
                strAtmtName(1) = Right$(strAtmtFullName,
Len(strAtmtFullName) - intDotPosition)

                ' Get the full saving path of the current attachment.
                strAtmtPath = strFolderPath & atmt.FileName

                ' /* If the length of the saving path is not larger than
260 characters.*/

                If Len(strAtmtPath) <= MAX_PATH Then
                    ' True: This attachment can be saved.
                    blnIsSave = True

                    ' /* Loop until getting the file name which does not
exist in the folder. */

                    Do While objFSO.FileExists(strAtmtPath)
                        strAtmtNameTemp = strAtmtName(0) &
Format(Now, "mmddhhmmss") & _
Format(Timer * 1000 Mod 1000,
"000")

                        strAtmtPath = strFolderPath & strAtmtNameTemp &
"." & strAtmtName(1)

                        ' /* If the length of the saving path is over 260
characters.*/

                        If Len(strAtmtPath) > MAX_PATH Then
                            lCountEachItem = lCountEachItem - 1
                            ' False: This attachment cannot be saved.
                            blnIsSave = False
                            Exit Do
                        End If
                    Loop

                    ' /* Save the current attachment if it is a valid file
name. */

                    If blnIsSave Then atmt.SaveAsFile strAtmtPath
                Else

```

```

        lCountEachItem = lCountEachItem - 1
    End If
    Next
End If

    ' Count the number of attachments in all Outlook items.
    lCountAllItems = lCountAllItems + lCountEachItem
Next
End If
Else
    MsgBox "Failed to get the handle of Outlook window!", vbCritical, "Error
from Attachment Saver"
    blnIsEnd = True
    GoTo PROC_EXIT
End If

' /* For run-time error:
'   The Explorer has been closed and cannot be used for further operations.
'   Review your code and restart Outlook. */
Else
    MsgBox "Please select an Outlook item at least.", vbExclamation, "Message from
Attachment Saver"
    blnIsEnd = True
End If

PROC_EXIT:
    SaveAttachmentsFromSelection = lCountAllItems

    ' /* Release memory. */
    If Not (objFSO Is Nothing) Then Set objFSO = Nothing
    If Not (objItem Is Nothing) Then Set objItem = Nothing
    If Not (selItems Is Nothing) Then Set selItems = Nothing
    If Not (atmt Is Nothing) Then Set atmt = Nothing
    If Not (atmts Is Nothing) Then Set atmts = Nothing

    ' /* End all code execution if the value of blnIsEnd is True. */
    If blnIsEnd Then End
End Function

' #####
' Convert general path.
' #####
Public Function CGPath(ByVal Path As String) As String
    If Right(Path, 1) <> "\" Then Path = Path & "\"
    CGPath = Path
End Function

' #####
' Run this macro for saving attachments.
' #####
Public Sub ExecuteSaving()
    Dim lNum As Long

    lNum = SaveAttachmentsFromSelection

    If lNum > 0 Then
        MsgBox CStr(lNum) & " attachment(s) was(were) saved successfully.",
vbInformation, "Message from Attachment Saver"
    Else
        MsgBox "No attachment(s) in the selected Outlook items.", vbInformation,
"Message from Attachment Saver"
    End If
End Sub

```

```
End If  
End Sub
```