



AddinManager.xlam
www.github.com/alexofrhodes

--- Table Of Contents ---

(Document)	ThisWorkbook
(Document)	Sheet1 - Sheet1
(Document)	Sheet2 - Sheet2
(Module)	Module1
(Module)	Module2
(Module)	vbArcImports
(UserForm)	uAddinManager
(UserForm)	uDEV
(UserForm)	vbArcAddinsForm

--- ThisWorkbook ---

--- Sheet2 - Sheet2 ---

--- Module1 ---

```
Sub AddinManagerButtonClicked(Control As IRibbonControl)
    uAddinManager.Show
End Sub
```

```
Public Function LastModified(filespec)
    Dim fs, f, s
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set f = fs.GetFile(filespec)
    LastModified = f.DateLastModified
End Function
```

```
Sub Admin_AddinsModified()
    On Error Resume Next
    Application.CalculateFull
    On Error GoTo 0
    Dim fso As Object
    Set fso = CreateObject("Scripting.FileSystemObject")
    Dim cell As Range
    Set cell = Sheet1.Range("E2")
    Dim ans As Long
    Do While cell <> ""
        If cell.Text = "A" Then
            ans = MsgBox("Replace" & vbTab & Sheet1.Range("A1") & cell. _
                Offset(0, -4) & vbTab & cell.Offset(0, -2) & vbNewLine & " _
                With" & vbTab & _
                Sheet1.Range("G1") & cell.Offset(0, 2) & vbTab & cell. _
                Offset(0, 4), vbYesNo, "Checking for updates")
            ElseIf cell.Text = "B" Then
                ans = MsgBox("Replace" & vbTab & Sheet1.Range("G1") & cell. _
                    Offset(0, 2) & vbTab & cell.Offset(0, 4) & vbNewLine & " _
                    With" & vbTab & _
                    Sheet1.Range("A1") & cell.Offset(0, -4) & vbTab & cell. _
                    Offset(0, -2), vbYesNo, "Checking for updates")
            End If

            If ans = vbYes Then
                If cell.Text = "A" Then
                    Workbooks(cell.Offset(0, 2).Text).IsAddin = True
                    Workbooks(cell.Offset(0, 2).Text).Close
                    fso.CopyFile cell.Offset(0, 3).Text, cell.Offset(0, -3). _
                        Text, True
                    Workbooks.Open cell.Offset(0, -3)
                ElseIf cell.Text = "B" Then
                    fso.CopyFile cell.Offset(0, -3).Text, cell.Offset(0, 3). _
                        Text, True
                End If
            End If
            Set cell = cell.Offset(1, 0)
        Loop
    Application.CalculateFull
End Sub
```

```

Sub FollowLink(FolderPath As String)
    Dim oShell As Object
    Dim Wnd As Object
    Set oShell = CreateObject("Shell.Application")
    For Each Wnd In oShell.Windows
        If Wnd.Name = "File Explorer" Then
            If Wnd.Document.Folder.Path = FolderPath Then Exit Sub
        End If
    Next Wnd
    Application.ThisWorkbook.FollowHyperlink Address:=FolderPath, _
    NewWindow:=True
End Sub

```

```

Function WorkbookIsOpen(ByVal swbkName As String) As Boolean
    WorkbookIsOpen = False
    On Error Resume Next
    WorkbookIsOpen = Len(Workbooks(swbkName).Name) <> 0
    On Error GoTo 0
End Function

```

--- Module2 ---

```
Private Declare PtrSafe Function URLDownloadToFile Lib "urlmon" _
Alias "URLDownloadToFileA" (ByVal pCaller As Long, ByVal szURL As String, _
_
ByVal szFileName As String, ByVal dwReserved As Long, ByVal lpfnCB As _
Long) As Long

Sub DownloadFile(FileUrl As String, SaveAs As String)
    URLDownloadToFile 0, FileUrl, SaveAs, 0, 0
End Sub

Function vbArcAddins() As Variant
    '#IMPORTS DownloadTextFile
    Dim v
    v = Filter(Split(DownloadTextFile("https://github.com/alexofrhodes/_
vbArc-addins/raw/main/ListOfAddins.txt"), " " & vbLf), "xlam", True)
    Dim i As Long
    vbArcAddins = v
End Function

Public Function FileExists(ByVal FileName As String) As Boolean
    If Right(FileName, 1) = "\" Then FileName = Left(FileName, _
Len(FileName) - 1)
    FileExists = (Dir(FileName, vbArchive + vbHidden + vbReadOnly + _
vbSystem) <> "")
End Function

Sub DownloadFileFromURL()
    '#IMPORTS DownloadFile

    Dim FileUrl As String
    Dim objXmlHttpReq As Object
    Dim objStream As Object
    FileUrl = "https://github.com/alexofrhodes/vbArc-addins/raw/main/Code _
Printer/CodePrinter.xlam"

    Set objXmlHttpReq = CreateObject("Microsoft.XMLHTTP")
    objXmlHttpReq.Open "GET", FileUrl, False, "username", "password"
    objXmlHttpReq.send

    If objXmlHttpReq.Status = 200 Then
        Set objStream = CreateObject("ADODB.Stream")
        objStream.Open
        objStream.Type = 1
        objStream.Write objXmlHttpReq.responseBody
        objStream.SaveToFile "C:\Users\acer\Downloads\CodePrinter.xlam", 2
        objStream.Close
    End If
End Sub
```



```

Function DownloadTextFile(URL As String) As String
    On Error GoTo Err_GetFromWebpage
    Dim objWeb As Object
    Dim strXML As String
    Set objWeb = CreateObject("Msxml2.ServerXMLHTTP")
    objWeb.Open "GET", URL, False
    objWeb.setRequestHeader "Content-Type", "text/xml"
    objWeb.setRequestHeader "Cache-Control", "no-cache"
    objWeb.setRequestHeader "Pragma", "no-cache"
    objWeb.send
    strXML = objWeb.responseText
    DownloadTextFile = strXML
End_GetFromWebpage:
    Set objWeb = Nothing
    Exit Function
Err_GetFromWebpage:
    MsgBox Err.Description & " (" & Err.Number & ")"
    Resume End_GetFromWebpage
End Function

```

--- vbArcImports ---

Rem -----

Rem Debug Print

Sub dp(Var As Variant)

'#IMPORTS ArrayDimensions

'#IMPORTS DPH

Dim element As Variant

Select Case TypeName(Var)

Case Is = "String", "Long", "Integer", "Boolean"

Debug.Print Var

Rem todo How to handle multidimensional array?

Case Is = "Variant()", "String()", "Long()", "Integer()"

If ArrayDimensions(Var) = 1 Then

Dim i As Long

For i = LBound(Var) To UBound(Var)

Debug.Print Var(i)

Next i

ElseIf ArrayDimensions(Var) > 1 Then

DPH Var

End If

Case Is = "Collection"

For Each element In Var

dp element

Next element

Case Else

End Select

End Sub

Function TextDecomposition(Mojiretu\$)

Dim i&, n&

Dim output

n = Len(Mojiretu)

ReDim output(1 To n)

For i = 1 To n

output(i) = Mid(Mojiretu, i, 1)

Next i

TextDecomposition = output

End Function

Function CalculateByteCharacters(Mojiretu\$)

Dim MojiKosu%

MojiKosu = Len(Mojiretu)

Dim output

ReDim output(1 To MojiKosu)

Dim i&

Dim TmpMoji\$

For i = 1 To MojiKosu

TmpMoji = Mid(Mojiretu, i, 1)

If i = 1 Then

output(i) = LenB(StrConv(TmpMoji, vbFromUnicode))

```

Else
    output(i) = LenB(StrConv(TmpMoji, vbFromUnicode)) + output(i - 1)
End If
Next i
CalculateByteCharacters = output
End Function

```

```

Function ShortenToByteCharacters(Mojiretu$, ByteNum%)
'#IMPORTS TextDecomposition
'#IMPORTS CalculateByteCharacters
Dim OriginByte%
Dim output
OriginByte = LenB(StrConv(Mojiretu, vbFromUnicode))
If OriginByte <= ByteNum Then
    output = Mojiretu
Else
    Dim RuikeiByteList, BunkaiMojiretu
    RuikeiByteList = CalculateByteCharacters(Mojiretu)
    BunkaiMojiretu = TextDecomposition(Mojiretu)
    Dim AddMoji$
    AddMoji = "."
    Dim i&, n&
    n = Len(Mojiretu)
    For i = 1 To n
        If RuikeiByteList(i) < ByteNum Then
            output = output & BunkaiMojiretu(i)
        ElseIf RuikeiByteList(i) = ByteNum Then
            If LenB(StrConv(BunkaiMojiretu(i), vbFromUnicode)) = 1 Then
                output = output & AddMoji
            Else
                output = output & AddMoji & AddMoji
            End If
            Exit For
        ElseIf RuikeiByteList(i) > ByteNum Then
            output = output & AddMoji
            Exit For
        End If
    Next i
End If
ShortenToByteCharacters = output
End Function

```

```

Sub DebugPrintHairetu(ByVal Hairetu, Optional HyoujiMaxNagasa%, Optional HairetuName$)
'#IMPORTS ShortenToByteCharacters
Dim i&, j&, k&, m&, n&
Dim TateMin&, TateMax&, YokoMin&, YokoMax&
Dim WithTableHairetu
Dim NagasaList, MaxNagasaList
Dim NagasaOnajilist
Dim OutputList

```

```

Const SikiriMoji$ = "|"
Dim Jigen2%
On Error Resume Next
Jigen2 = UBound(Hairetu, 2)
On Error GoTo 0
If Jigen2 = 0 Then
    Hairetu = Application.Transpose(Hairetu)
End If
TateMin = LBound(Hairetu, 1)
TateMax = UBound(Hairetu, 1)
YokoMin = LBound(Hairetu, 2)
YokoMax = UBound(Hairetu, 2)
ReDim WithTableHairetu(1 To TateMax - TateMin + 1 + 1, 1 To YokoMax - _
YokoMin + 1 + 1)
For i = 1 To TateMax - TateMin + 1
    WithTableHairetu(i + 1, 1) = TateMin + i - 1
    For j = 1 To YokoMax - YokoMin + 1
        WithTableHairetu(1, j + 1) = YokoMin + j - 1
        WithTableHairetu(i + 1, j + 1) = Hairetu(i - 1 + TateMin, j - _
1 + YokoMin)
    Next j
Next i
n = UBound(WithTableHairetu, 1)
m = UBound(WithTableHairetu, 2)
ReDim NagasaList(1 To n, 1 To m)
ReDim MaxNagasaList(1 To m)
Dim TmpStr$
For j = 1 To m
    For i = 1 To n
        If j > 1 And HyoujiMaxNagasa <> 0 Then
            TmpStr = WithTableHairetu(i, j)
            WithTableHairetu(i, j) = ShortenToByteCharacters(TmpStr, _
HyoujiMaxNagasa)
        End If
        NagasaList(i, j) = LenB(StrConv(WithTableHairetu(i, j), _
vbFromUnicode))
        MaxNagasaList(j) = WorksheetFunction.Max(MaxNagasaList(j), _
NagasaList(i, j))
    Next i
Next j
ReDim NagasaOnajiliList(1 To n, 1 To m)
Dim TmpMaxNagasa&
For j = 1 To m
    TmpMaxNagasa = MaxNagasaList(j)
    For i = 1 To n
        NagasaOnajiliList(i, j) = WithTableHairetu(i, j) & _
WorksheetFunction.Rept(" ", TmpMaxNagasa - NagasaList(i, j))
    Next i
Next j
ReDim OutputList(1 To n)
For i = 1 To n
    For j = 1 To m
        If j = 1 Then

```

```

        OutputList(i) = NagasaOnajiList(i, j)
    Else
        OutputList(i) = OutputList(i) & SikiriMoji & _
        NagasaOnajiList(i, j)
    End If
Next j
Next i
Debug.Print HairetuName
For i = 1 To n
    Debug.Print OutputList(i)
Next i
End Sub

```

```

Function ArrayDimensions(ByVal vArray As Variant) As Long
    Dim dimnum As Long
    On Error GoTo FinalDimension
    For dimnum = 1 To 60000
        ErrorCheck = LBound(vArray, dimnum)
    Next
FinalDimension:
    ArrayDimensions = dimnum - 1
End Function

```

```

Rem -----

```

```

Sub DPH(ByVal Hairetu, Optional HyoujiMaxNagasa%, Optional HairetuName$)
    '#IMPORTS DebugPrintHairetu
    Call DebugPrintHairetu(Hairetu, HyoujiMaxNagasa, HairetuName)
End Sub

```

--- uAddinManager ---

```
Private Sub CommandButton1_Click()
```

```
For i = 0 To body.ListCount - 1
```

```
    If body.Selected(i) = True Then AddIns(body.List(i, 0)).Installed = _
```

```
    Not AddIns(body.List(i, 0)).Installed
```

```
Next
```

```
LoadAddins
```

```
End Sub
```

```
Private Sub CommandButton2_Click()
```

```
Dim ans As Long
```

```
ans = MsgBox("Irreversible. Proceed?", vbYesNo)
```

```
If ans = vbNo Then Exit Sub
```

```
AddIns(body.List(body.ListIndex, 0)).Installed = False
```

```
Kill AddIns(body.List(body.ListIndex, 0)).FullName
```

```
End Sub
```

```
Private Sub CommandButton3_Click()
```

```
SortListboxOnColumn body, 0
```

```
End Sub
```

```
Private Sub CommandButton4_Click()
```

```
SortListboxOnColumn body, 1
```

```
End Sub
```

```
Private Sub CommandButton5_Click()
```

```
vbArcAddinsForm.Show
```

```
Unload Me
```

```
End Sub
```

```
Private Sub info_MouseDown(ByVal Button As Integer, ByVal Shift As _
```

```
Integer, ByVal X As Single, ByVal Y As Single)
```

```
uDEV.Show
```

```
End Sub
```

```
Private Sub Label1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
```

```
Admin_AddinsModified
```

```
End Sub
```

```
Private Sub UserForm_Initialize()
```

```
LoadAddins
```

```
End Sub
```

```
Private Sub LoadAddins()
```

```
'#IMPORTS SortListboxOnColumn
```

```
body.Clear
```

```
Dim ad As AddIn
```

```
On Error Resume Next
```

```
For Each ad In AddIns
```

```
    body.AddItem
```

```
    body.List(body.ListCount - 1, 0) = Left(ad.Name, InStr(1, ad.Name, ". _
```

```
    ") - 1)
```

```
body.List(body.ListCount - 1, 1) = IIf(ad.Installed, " ENABLED", "-")
```

```
Next
```

```
SortListboxOnColumn body, 1
```

```
End Sub
```

```
Sub SortListboxOnColumn(lBox As MSForms.ListBox, Optional OnColumn As _  
Long = 0)
```

```
Dim vntData As Variant
```

```
Dim vntTempItem As Variant
```

```
Dim lngOuterIndex As Long
```

```
Dim lngInnerIndex As Long
```

```
Dim lngSubItemIndex As Long
```

```
vntData = lBox.List
```

```
For lngOuterIndex = LBound(vntData, 1) To UBound(vntData, 1) - 1
```

```
For lngInnerIndex = lngOuterIndex + 1 To UBound(vntData, 1)
```

```
If vntData(lngOuterIndex, OnColumn) > vntData(lngInnerIndex, _  
OnColumn) Then
```

```
For lngSubItemIndex = 0 To lBox.ColumnCount - 1
```

```
vntTempItem = vntData(lngOuterIndex, lngSubItemIndex)
```

```
vntData(lngOuterIndex, lngSubItemIndex) = _
```

```
vntData(lngInnerIndex, lngSubItemIndex)
```

```
vntData(lngInnerIndex, lngSubItemIndex) = vntTempItem
```

```
Next
```

```
End If
```

```
Next lngInnerIndex
```

```
Next lngOuterIndex
```

```
lBox.Clear
```

```
lBox.List = vntData
```

```
End Sub
```

--- uDEV ---

```
Private Sub LFaceBook_Click()
FollowLink ("https://www.facebook.com/VBA-Code-Archive-110295994460212")
End Sub

Private Sub LGitHub_Click()
FollowLink ("https://github.com/alexofrhodes")
End Sub

Private Sub LYouTube_Click()
FollowLink ("https://bit.ly/2QT4wFe")
End Sub

Private Sub LBuyMeACoffee_Click()
FollowLink ("http://paypal.me/alexofrhodes")
End Sub

Private Sub FollowLink(FolderPath As String)
Dim oShell As Object
Dim Wnd As Object
Set oShell = CreateObject("Shell.Application")
For Each Wnd In oShell.Windows
    If Wnd.Name = "File Explorer" Then
        If Wnd.Document.Folder.Self.Path = FolderPath Then Exit Sub
    End If
Next Wnd
Application.ThisWorkbook.FollowHyperlink Address:=FolderPath, _
NewWindow:=True
End Sub

Private Sub LEmail_Click()
If OutlookCheck = True Then
    MailDev
Else
    Dim out As String
    out = "anastasioualex@gmail.com"
    CLIP out
    MsgBox ("Outlook not found" & Chr(10) & _
    "DEV's email address" & vbNewLine & out & vbNewLine & "copied to _
    clipboard")
End If
End Sub

Sub MailDev()
Dim OutApp As Object
Dim OutMail As Object
Dim strBody As String
Set OutApp = CreateObject("Outlook.Application")
Set OutMail = OutApp.CreateItem(0)
On Error Resume Next
With OutMail
    .To = "anastasioualex@gmail.com"
```



```

        .CC = vbNullString
        .BCC = vbNullString
        .Subject = "DEV REQUEST OR FEEDBACK FOR -CODE ARCHIVE-"
        .body = strBody
        .Display
    End With
    On Error GoTo 0
    Set OutMail = Nothing
    Set OutApp = Nothing
End Sub

```

```

Private Function CLIP(Optional StoreText As String) As String
Dim X As Variant
X = StoreText
With CreateObject("htmlfile")
    With .parentWindow.clipboardData
        Select Case True
        Case Len(StoreText)
            .SetData "text", X
        Case Else
            CLIP = .GetData("text")
        End Select
    End With
End With
End Function

```

--- vbArcAddinsForm ---

```
Private Sub CommandButton1_Click()
```

```
DownloadThis
```

```
End Sub
```

```
Private Sub CommandButton2_Click()
```

```
For i = 0 To ListBox1.ListCount - 1
```

```
    If ListBox1.Selected(i) = True Then
```

```
        If ListBox1.List(i, 1) = "INSTALLED" Then
```

```
            AddinName = Left(ListBox1.List(i, 0), InStrRev(ListBox1. _
```

```
                List(i, 0), ".") - 1)
```

```
            AddIns(AddinName).Installed = Not AddIns(AddinName).Installed
```

```
            ListBox1.List(i, 2) = IIf(AddIns(AddinName).Installed = True, _
```

```
                "ENABLED", "DISABLED")
```

```
        End If
```

```
    End If
```

```
Next
```

```
End Sub
```

```
Private Sub info_MouseDown(ByVal Button As Integer, ByVal Shift As _
```

```
Integer, ByVal X As Single, ByVal Y As Single)
```

```
uDEV.Show
```

```
End Sub
```

```
Private Sub UserForm_Initialize()
```

```
Dim v As Variant
```

```
v = vbArcAddins
```

```
Dim i As Long
```

```
On Error Resume Next
```

```
For i = 0 To UBound(v)
```

```
    ListBox1.AddItem
```

```
    ListBox1.List(i, 0) = Mid(v(i), InStrRev(v(i), "/") + 1)
```

```
    Debug.Print ListBox1.List(i, 0)
```

```
    ListBox1.List(i, 1) = IIf(FileExists(Application.UserLibraryPath & _
```

```
        ListBox1.List(i, 0)) = False, "MISSING", "INSTALLED")
```

```
    AddinName = Left(ListBox1.List(i, 0), InStrRev(ListBox1.List(i, 0), ". _
```

```
        ") - 1)
```

```
    Set ad = AddIns(AddinName)
```

```
    If Not ad Is Nothing Then
```

```
        ListBox1.List(i, 2) = IIf(AddIns(AddinName).Installed = True, _
```

```
            "ENABLED", "DISABLED")
```

```
    End If
```

```
    On Error GoTo 0
```

```
    ListBox1.List(i, 3) = v(i)
```

```
Next
```

```
End Sub
```

```
Sub DownloadThis()
```

```
'#IMPORTS DownloadFile
```

```

Dim i As Long
Dim AddinName As String
Dim ad As AddIn

For i = 0 To ListBox1.ListCount - 1
    If ListBox1.Selected(i) = True Then
        AddinName = Left(ListBox1.List(i, 0), InStrRev(ListBox1. _
            List(i, 0), ".") - 1)
        On Error Resume Next
        Set ad = AddIns(AddinName)
        If Not ad Is Nothing Then
            If AddIns(AddinName).Installed = True Then _
                AddIns(AddinName).Installed = False
        End If

        Kill Application.UserLibraryPath & ListBox1.List(i, 0)
        On Error GoTo 0
        DownloadFile ListBox1.List(i, 3), Application.UserLibraryPath & _
            & ListBox1.List(i, 0)
        AddIns.Add (Application.UserLibraryPath & ListBox1.List(i, 0))
        AddIns(AddinName).Installed = True
        ListBox1.List(i, 1) = "INSTALLED"
        ListBox1.List(i, 2) = "ENABLED"
    End If
Next
End Sub

```