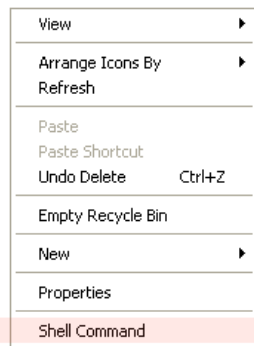
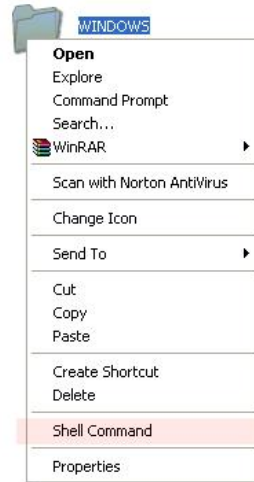


Create Shell Extensions ContextMenu Handler

اضافة عنصر قائمة الي قائمة مستعرض الويندوز

من خلال هذا الموضوع سنوضح كيف يمكن اضافة عنصر قائمة الي قائمة مستعرض الويندوز، عند الضغط كليك يمين علي الملف او المجلد.



نبدأ المشروع بسم الله

اولاً : ننشي ملف ونسميه ShellExtLib.vb ونكتب بداخله الاكواد التالية

فيما يلي الـ Interface

والخاص بأستيراد مقبض قائمة ContextMenu عند الضغط كليك يمين علي الملف او المجلد او في خلفية المجلد بمستعرض الويندوز.

```
< _
    ComImport(), _
    ' قيمتها كما في الرجستري
    Guid("000214E4-0000-0000-C000-000000000046"), _

    InterfaceType(ComInterfaceType.InterfaceIsIUnknown) _
> _
Friend Interface IContextMenu

    ' مسؤلة عن ايجاد مقبض القائمة وازافة عنصر قائمة
    <PreserveSig(>> _
    Function QueryContextMenu( _
        ByVal hMenu As IntPtr, _
        ByVal indexMenu As Integer, _
        ByVal idCmdFirst As Integer, _
        ByVal idCmdLast As Integer, _
        ByVal uFlags As CMF) As Integer

    ' مسؤلة عن الضغط بالماوس علي عنصر القائمة
    Sub InvokeCommand( _
        ByRef lpici As CMINVOKECOMMANDINFO)

    ' مسؤلة عن كتابة بيانات في شريط الحالة
    Sub GetCommandString( _
        ByVal idCmd As Integer, _
        ByVal uType As GetCommandStringFlags, _
        ByVal pwReserved As Integer, _
        ByVal pszName As IntPtr, _
        ByVal cchMax As Integer)

End Interface
```

```

<Flags()> _
Friend Enum MF
    Append                = &H100
    Bitmap                = &H4
    ByCommand             = &H0
    ByPosition            = &H400
    Change                = &H80
    Checked               = &H8
    [Default]             = &H1000
    Delete                = &H200
    Disabled              = &H2
    Enabled               = &H0
    [End]                 = &H80
    Grayed               = &H1
    Help                  = &H4000
    Hilite                = &H80
    Insert                = &H0
    MenuBarBreak          = &H20
    MenuBreak             = &H40
    MouseSelect           = &H8000
    OwnerDraw             = &H100
    PopUp                 = &H10
    Remove                = &H1000
    RightJustify          = &H4000
    Separator             = &H800
    [String]              = &H0
    SysMenu               = &H2000
    UnChecked             = &H0
    UnHilite              = &H0
    UseCheckBitmaps       = &H200
End Enum

' QueryContextMenuFlags
<Flags()> _
Friend Enum CMF
    NORMAL                = &H0
    DEFAULTONLY           = &H1
    VERBSONLY             = &H2
    EXPLORE               = &H4
    NOVERBS               = &H8
    CANRENAME             = &H10
    NODEFAULT            = &H20
    INCLUDESTATIC         = &H40
    RESERVED              = &HFFFFFF0000
End Enum

' GetCommandStringFlags
<Flags()> _
Friend Enum GCS
    VERB                  = &H0
    HELPTEXT              = &H1
    VALIDATE              = &H2
    [UNICODE]             = &H4
End Enum

' InvokeCommand
<Flags()> _
Friend Enum InvokeCommandMask
    CMIC_MASK_HOTKEY      = &H20
    CMIC_MASK_ICON        = &H10
    CMIC_MASK_FLAG_NO_UI  = &H400
    CMIC_MASK_UNICODE     = &H4000
    CMIC_MASK_NO_CONSOLE  = &H8000
    CMIC_MASK_ASYNCOK     = &H100000
    CMIC_MASK_PTINVOKE    = &H20000000
    CMIC_MASK_SHIFT_DOWN  = &H10000000
    CMIC_MASK_CONTROL_DOWN = &H20000000
End Enum

<EditorBrowsable(EditorBrowsableState.Never)> _
Friend Structure CMINVOKECOMMANDINFO
    Dim cbSize As Integer
    Dim fMask As InvokeCommandMask
    Dim hwnd As IntPtr
    Dim lpVerb As Integer
    Dim lpParameters As IntPtr
    Dim lpDirectory As IntPtr
    Dim nShow As Integer
    Dim dwHotKey As Integer
    Dim hIcon As IntPtr
End Structure

```

الى Interface التالي IShellExtInit خاص بارسال اسماء الملفات والمجلدات التي تم تحديدهما من قبل المستخدم قبل انشاء القوائم

```
< _
    ComImport(), _
    Guid("000214E8-0000-0000-C000-000000000046"), _
    InterfaceType(ComInterfaceType.InterfaceIsIUnknown) _
> _
Friend Interface IShellExtInit

    Sub Initialize( _
        ByVal pidlFolder As IntPtr, _

        <MarshalAs(UnmanagedType.Interface)> _
        ByVal lpIDataObject As IDataObject, _

        ByVal hkeyProgID As Integer)

End Interface
```

ويستخدم معها الـ Flags و Structure التالية اذا كنت تستخدم اصدار 2.0.NET. فقد توفر هذا الانترفايس IDataObject في فضاء الاسماء التالي System.Runtime.InteropServices.ComTypes

```
< _
    ComImport(), _
    Guid("0000010e-0000-0000-C000-000000000046"), _
    InterfaceType(ComInterfaceType.InterfaceIsIUnknown) _
> _
Friend Interface IDataObject
    Sub GetData(ByRef pformatetc In As FORMATETC, ByRef pmedium As STGMEDIUM)
    Sub GetDataHere(ByRef pformatetc As FORMATETC, ByRef pmedium As STGMEDIUM)
    <PreserveSig()> _
    Function QueryGetData(ByRef pformatetc As FORMATETC) As Integer
    Sub GetCanonicalFormatEtc(ByRef pformatetc In As FORMATETC, ByRef pformatetcOut As FORMATETC)
    Sub SetData(ByRef pformatetc As FORMATETC, ByRef pmedium As STGMEDIUM, _
        <MarshalAs(UnmanagedType.Bool)> ByVal fRelease As Boolean)
    Function EnumFormatEtc(ByVal dwDirection As DATADIR) As <MarshalAs(UnmanagedType.Interface)> Object
    Function DAdvise(ByRef pformatetc As FORMATETC, ByVal advf As Integer, _
        <MarshalAs(UnmanagedType.Interface)> ByVal pAdvSink As Object) As Integer
    Sub DUnadvise(ByVal dwConnection As Integer)

    Function EnumDAdvise() As <MarshalAs(UnmanagedType.Interface)> Object

End Interface

<Flags()> _
Friend Enum DATADIR
    DATADIR_GET = 1
    DATADIR_SET = 2
End Enum

<Flags()> _
Friend Enum TYMED
    TYMED_HGLOBAL = 1
    TYMED_FILE = 2
    TYMED_ISTREAM = 4
    TYMED_IStorage = 8
    TYMED_GDI = 16
    TYMED_MFPICT = 32
    TYMED_ENHMF = 64
    TYMED_NULL = 0
End Enum

<Flags()> _
Friend Enum DVASPECT
    DVASPECT_CONTENT = 1
    DVASPECT_THUMBAIL = 2
    DVASPECT_ICON = 4
    DVASPECT_DOCPRINT = 8
End Enum

<EditorBrowsable(EditorBrowsableState.Never)> _
Friend Structure FORMATETC
    Dim cfFormat As Short
    Dim ptd As Integer
    Dim dwAspect As DVASPECT
    Dim lindex As Integer

    Dim tymed As TYMED
End Structure

<EditorBrowsable(EditorBrowsableState.Never)> _
Friend Structure STGMEDIUM
    Dim tymed As TYMED
    Dim data As Integer
    <MarshalAs(UnmanagedType.IUnknown)> _
    Dim punkForRelease As Object
End Structure
```

```
Public Class DllImport

    ' ContextMenu عن اضافة عنصر قائمة الي قائمة
    Friend Declare Ansi Function InsertMenu Lib "User32" Alias "InsertMenuA" ( _
        ByVal hMenu As IntPtr, _
        ByVal nPosition As Integer, _
        ByVal wFlags As MF, _
        ByVal wIDNewItem As Integer, _
        ByVal lpNewItem As String _
    ) As Long

    ' مسؤولية عن ايجاد الملفات والجلدات التي تم تحديدها
    Friend Declare Ansi Function DragQueryFile Lib "shell32" Alias "DragQueryFileA" ( _
        ByVal hDrop As Integer, _
        ByVal iFile As Integer, _
        ByVal lpszFile As StringBuilder, _
        ByVal cch As Integer _
    ) As Integer

    ' مسؤولية عن ايجاد مسار الجلد عند الضغط كلك عين في خلفية الجلد
    Friend Declare Function SHGetPathFromIDList Lib "shell32" ( _
        ByVal pidl As IntPtr, _
        ByVal path As StringBuilder _
    ) As Integer

    Friend Declare Function ReleaseStgMedium Lib "ole32" (ByRef Stm As STGMEDIUM) As Integer

End Class
```

ويتم وضع فضاء الاسماء التالية في رأس الملف.

```
Imports System
Imports System.Text
Imports System.ComponentModel
Imports System.Runtime.InteropServices

Namespace ShellExt.Interop

    ' الانتفايس تكتب هنا

End Namespace
```

الان انتهينا من كتابة الانتفايس وونتقل الي استخدا مها .

ثانياً : ننشئ ملف جديد باسم ShellCommand.vb ونكتب بداخله فضاء الاسماء التالية واسم فضاء الاسماء الخاص بالمشروع واسم الكلاس.

```
Option Strict On
Option Explicit On

Imports System
Imports System.Collections
Imports System.ComponentModel
Imports System.Drawing
Imports System.Text
Imports System.Text.Encoding
Imports System.Runtime.InteropServices
Imports System.Windows.Forms
Imports Microsoft.Win32
Imports Microsoft.VisualBasic

' فضاء الاسماء الذي قمنا بعمله سابقاً
Imports ShellExt.Interop

' فضاء اسماء الخاص بالمشروع
Namespace ShellExt

    ' اضافة خاصية اظهار الكلاس لكائنات COM
    <ComVisible(True)> _

    ' Guid نضيف الصفة التالية
    ' Visual Studio -> Tools -> Create GUID هذه القيمة من قائمة
    ' نتم اختيار الخيار الرابع واضغط علي زر نسخ ونلصق ونحذف القوسين من الاطراف
    <Guid("ACD2EEF8-5B96-4d8b-9F2F-2A514F050B4F")> _

    ' ShellCommand عن انشاء كلاس جديد باسم
    Public Class ShellCommand
        ' نستورد الانترفايس التالية من ShellExt.Interop ومن ثم نورثها لكلاسنا
        Implements IShellExtInit
        Implements IContextMenu

        ' نقوم بالتصريح عن مصفوفة من نوع ArrayList ليتم نسخ مسارات الملفات والمجلدات اليها
        Dim list as New ArrayList

        ' ننشئ الثابت التالي وهو من نوع نصي ويأخذ مساراً في الرجستري
        Const shell_Approved as string = _
            "Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved"

        ' كل نوع من الملفات يحتوي علي هذه القيمة في الرجستري
        Const root as string = "\shellex\ContextMenuHandlers\"

        ' وهذا الثابت يأخذ قيمة Guid الذي كتبناه في صفات الكلاس ونضيف له القوسين في الاطراف
        const ClassId as string = "{ACD2EEF8-5B96-4d8b-9F2F-2A514F050B4F}"

    End Class

End Namespace
```

بعد ان قمنا بانشاء الملف واضفنا اليه الكلاس واضفنا له بعض الصفات "Attributes"،

سنقوم بأضافة اول اجراء "Initialize" والمشتق من انترفاس IShellExtInit والمسؤل عن ايجاد اسماء الملفات والمجلدات التي قد تم تحديدها من قبل المستخدم وتتم هذا المهمة قبل انشاء القائمة لكي تضيف بعض الاجراءات المستقبلية مثلاً كتحديد نوعية الملف وعدم اظهارها اذا كان نوع الملف غير مرغوب به ويكتب الاجراء هكذا.

```
'اجراء ايجاد مسارات الملفات والمجلدات التي تم تحديدهما قبل انشاء القائمة'
Private Sub Initialize( _
    ByVal pidlFolder As IntPtr, _
    ByVal lpIDataObject As ShellExt.Interop.IDataObject, _
    ByVal hkeyProgID As Integer) Implements IShellExtInit.Initialize

    Try
        'تحديد ما اذا كان الضغط في خلفية المجلد من باريمتر pidlFolder
        If Not pidlFolder.Equals(IntPtr.Zero) Then

            'انشاء متغير من نوع StringBuilder لكي يتم نسخ المسار اليه'
            Dim sb As New StringBuilder(260)

            'نسخ مسار المجلد الحالي من خلال الـ API التالي'
            DllImport.SHGetPathFromIDList(pidlFolder, sb)

            'اضافة المسار المسوخ الي المصفوفة'
            list.Add(sb.ToString)

            'تحرير الذاكرة من متغير sb'
            sb = Nothing

            'اذا كان هناك ملفات او مجلدات قد تم تحديدهما'
            ElseIf Not lpIDataObject Is Nothing Then

                Dim FMT As New FORMATETC
                Dim STM As New STGMEDIUM

                With FMT
                    .cfFormat = 15 ' CF_HDROP
                    .tymed = TYMED.TYMED_HGLOBAL
                    .dwAspect = DVASPECT.DVASPECT_CONTENT
                End With

                Try
                    Dim sb As StringBuilder
                    lpIDataObject.GetData(FMT, STM)

                    'ايجاد عدد الملفات او المجلدات المحددة'
                    Dim filesCount As Integer = DllImport.DragQueryFile(STM.data, -1, Nothing, 0)
                    'انشاء جملة تكرارية لنسخ جميع اسماء الملفات'
                    For I As Integer = 0 To filesCount - 1
                        sb = New StringBuilder(260)
                        'نسخ مسارات الملفات والمجلدات المحددة'
                        DllImport.DragQueryFile(STM.data, I, sb, sb.Capacity)
                        'list اضافة المسار الحالي الي مصفوفة list'
                        list.Add(sb.ToString)
                    Next
                    'تحرير الذاكرة من متغير sb'
                    sb = Nothing
                Finally
                    'تحرير الذاكرة من متغير STM'
                    DllImport.ReleaseStgMedium(STM)
                End Try
            End If

            Catch ex As Exception
            End Try

        End Sub
```

الاجراء التالي QueryContextMenu والمشتق من انترفاس IContextMenu والمسؤل عن ايجاد مقبض القائمة،
واضافة عنصر قائمة مع خصائص ويكتب الاجراء كالتالي :

```
' ايجاد مقبض القائمة ووضع عنصر بها
Private Function QueryContextMenu( _
    ByVal hMenu As IntPtr, _ ' يعيد مقبض القائمة المستهدفة
    ByVal indexMenu As Integer, _ ' رقم ترتيب عنصر القائمة دائماً يساوي صفر
    ByVal idCmdFirst As Integer, _ ' يعيد قيمة العنصر المضاف ليستخدم في الاجراءات الاخرى
    ByVal idCmdLast As Integer, _
    ByVal uFlags As CMF) As Integer Implements IContextMenu.QueryContextMenu

    ' idCmdFirst نصرح عن متغير ونعطيه قيمة
    dim id As Integer = idCmdFirst

    ' اضافة عنصر القائمة مع بياناته من خلال API التالي
    DllImport.InsertMenu( _
        hMenu, _ ' مقبض القائمة
        indexMenu, _ ' رقم ترتيب العنصر داخل القائمة
        MF.String Or MF.ByCommand, _ ' خصائص العنصر المضاف وهي كثير ولكي اضفت تلك الخواص الثانية ترتيب العناصر حسب الامر
        idCmdFirst, _
        "Shell Command" _ ' اسم عنصر القائمة المضاف
    )

    ' نقوم باضافة رقم ١ لهذا المتغير في كل مرة فيها اضافة عنصر قائمة جديد
    idCmdFirst += 1

    ' نطرح id من الاعداد التي تم اضافتها الي متغير idCmdFirst ومن ثم نرسله للاجراء
    Return (idCmdFirst - id)

End Function
```

نأتي الي اجراء InvokeCommand والمشتق من انترفاس IContextMenu والمسؤل عن امر الضغط علي عنصر القائمة المضاف ويكتب هكذا.

```
' اجراء امر الضغط علي عنصر القائمة
Private Sub InvokeCommand(ByRef lpici As CMINVOKECOMMANDINFO) Implements IContextMenu.InvokeCommand

    ' المتغير lpici من نوع سترات قد قمنا بأنشائها في ملف ShellExtLib.vb
    ' ويحتوي علي متغير بأسم lpverb وهذا المتغير يشير الي عدد عناصر القائمة المضافة حسب الترتيب
    ' التأكد من ان هناك عناصر قائمة قد تم اضافتها
    IF lpici.lpVerb >= 0 Then
        dim s, sr as String

        ' وهذا حلقة تكرارية تأخذ مسارات الملفات والمجلدات التي تم نسخها مسبقاً وتعرضها في رسالة نصية
        For Each s in list
            sr &= s & vbNewLine
        Next

        MsgBox(sr)
    End IF

End Sub
```

نأتي الي اجراء GetCommandString والمشتق من انترفاس IContextMenu والمسؤل عن كتابة بيانات في شرط الحالة اسفل المستعرض.

```
' اجراء ارسال بيانات الي شريط الحالة اسفل المستعرض
Private Sub GetCommandString( _
    ByVal idCmd As Integer, _      ' رقم عنصر القائمة المضاف مسبقاً
    ByVal uType As GCS, _
    ByVal pwReserved As Integer, _
    ByVal pszName As IntPtr, _
    ByVal cchMax As Integer) Implements IContextMenu.GetCommandString

    Dim text As String = "Shell Command, Get Selected name(s)" & vbNullChar

    If (uType And GCS.HELPTTEXT) = GCS.HELPTTEXT Then

        If text.Length < cchMax Then cchMax = text.Length

        ' التحقق من المدخلات في المستعرض اما UNICODE او ANSI
        If (uType And GCS.UNICODE) = GCS.UNICODE Then
            Marshal.Copy(text.ToCharArray, 0, pszName, cchMax)
        Else
            Marshal.Copy(ASCII.GetBytes(text), 0, pszName, cchMax)
        End If
    End If
End Sub
```

الان تم الانتهاء من كتابة جميع اجراءات اضافة عنصر الي القائمة،

ولم يبق لنا إلا اجراء تسجيل وإلغاء الكلاس من الرجستري ويتم كتابة اجراء التسجيل هكذا.

```
<ComRegisterFunction()> _ ' يجب اضافة هذه الصفة حتي يتم التعرف علي هذا الاجراء اثناء تسجيل الكلاس
Shared Sub ComReg(ByVal t As Type)

    ' تسجيل الكلاس في مفتاح Shell Extensions
    Registry.LocalMachine.OpenSubKey( _
        shell_Approved, _ ' قد صرحنا عنه مسبقاً
        True).SetValue(
        ClassId, _ ' قد صرحنا عنه مسبقاً
        "My Shell Command")

    ' اضافة لواحق او امتداد الملفات التي نريد اظهار عنصر القائمة
    Registry.ClassesRoot.CreateSubKey( _

        ' نكتب هنا نوع الملف الذي نريد اضافة الكلاس اليه وهنا قمنا بأضافة ملفات ذو الامتداد txt
        "txtfile" & root + t.Name).SetValue("", ClassId)

    ' اضافة الكلاس الي قائمة خلفية الجلد
    Registry.ClassesRoot.CreateSubKey("Directory\Background" & root + t.Name).SetValue("", ClassId)
End Sub
```

اجراء إلغاء تسجيل الكلاس من الرجستري.

```
<ComUnRegisterFunction()> _ ' يجب اضافة هذه الصفة حتي يتم التعرف علي هذا الاجراء اثناء إلغاء تسجيل الكلاس
Shared Sub ComUnReg(ByVal t As Type)

    ' حذف المدخلات التي قمنا باضافتها في مسجل النظام الرجستري اثناء التسجيل
    Registry.LocalMachine.OpenSubKey(shell_Approved, true).DeleteValue(ClassId)
    Registry.ClassesRoot.DeleteSubKey("txtfile" & root + t.Name)
    Registry.ClassesRoot.DeleteSubKey("Directory\Background" & root + t.Name)
End Sub
```

الان انتهينا كلياً من كتابة المشروع.

ولم بقي الا ان نتعرف علي كيفية تسجيل وإلغاء التسجيل من داخل البرنامج او من خارجه.

اولا من خارج البرنامج ويكون بأمر RegAsm.exe وهو موجود في مجلد الفريم ورك
وطريقة كتابته التسجيل هكذا من نافذة Command Prompt

```
RegAsm <اسم الملف> /codebase  
التسجيل  
RegAsm ShellCommand.dll /codebase  
إلغاء التسجيل  
RegAsm ShellCommand.dll /unregister
```

ثانياً عن طريق الكود داخل البرنامج ويكتب هكذا

```
Dim regsv As New System.Runtime.InteropServices.RegistrationServices  
التسجيل  
regsv.RegisterAssembly(System.Reflection.Assembly.GetExecutingAssembly, AssemblyRegistrationFlags.SetCodeBase)  
إلغاء التسجيل  
regsv.UnregisterAssembly(System.Reflection. Assembly.GetExecutingAssembly)
```

في بعض حالات التسجيل وإلغاءة يحتاج الويندوز الي تسجيل خروج او اعاد تشغيله مرة اخرى.
والله الموفق