

بسم الله السميع العليم

سلسلة تقنيات تحسين واجهات البرامج

النوافذ التطبيقية

بقلم : رغيد الطيب

قيل قديما بأن إرضاء الناس غاية لا تدرك ! وهو امر حقيقي مسلم به في عيون كل الناس العادين ... إلا فئة معينة من الناس !! ... و هم المبرمجون ..

ه م الوحيدون الذين إذا سألتهم ... لقالوا إرضاء (المستخدم) غاية لا تدرك !! ... وذلك لكون الإنسان العادي يمكنه ان يقول ببساطة : "صحيح إرضاء الناس غاية لا تدرك" ويستريح!

غير ان المبرمج لا يمكنه ان يقول هذا ... لأنه إن فعل ... فلن يبقى هناك من يشتري برامجه !!

على العموم وبعيدا عن احزاني و شجوني البرمجية ... اقول يمكن للمبرمج الاعتماد الى الكثير من الحيل و الخدع و اللعب بالاكواد البرمجية لجذب إنتباه و إستحسان المستخدم (العادي) ... طبعا المستخدم العادي ... لأن المستخدم الخبير لن يلتفت الى جمال واجهاتك و تناسق ألوانك ... و لكنه سيلتفت الى المضمون وسهولة الاستخدام و هل حقق برنامجك غرضه الرئيسي ام لا.

و عليه فلا بأس ان نعتني بالاثنتين الشكل و المضمون ... و في هذه السلسلة ... التي اسميتها (تقنيات تحسين واجهات البرامج) سنناقش سويا بإذن الله تعالى بعض الحيل و الخدع (ما اقبح الكلمات و أنبل الغرض) ... اتمنى ان لا تفهم اننا في هذه السلسلة سرننتهي منها وقد اصبحنا محتالين و مخادعين و العياذ بالله

إن كنا سنخدع فإننا سنخدع الويندوز و النوافذ و الأدوات لجعلها تبدو اجمل في عيون مستخدمنا العزيز.

الخدعة الأولى

النوافذ التطبيقية

النافذة ببساطة هي الفورم العادية كا Form1 و هي نافذة عادية يمكن التعديل عليها ساعة إنشائها و جعلها تتصف بصفات معينة مثل شفافة, متحركة, ثابتة, لها عنوان ام لا ... وهكذا ...

ولكي خواص النوافذ توجد عدة طرق اما بإستخدام الخاصية CreateParams التابعة للفورم ويمكن ذلك ايضا عن طريق دوال ال API كالدالة SetWindowLong

و عن طريق اي الطرق سنقوم بتغيير خواص الفورم الى WS_LayeredWindow و هي ان نجعل الفورم عبارة عن فورم طبقية ... كالتالي :

```
Dim exStyle As Int32 = GetWindowLong(Me.Handle.ToInt32, GWL_EXSTYLE)
SetWindowLong(Me.Handle.ToInt32, GWL_EXSTYLE, exStyle Or WS_EX_LAYERED)
```

من الكود السابق ستعرف اني قمت باستخدام دوال الـ SetWindowLong و GetWindowLong ...

من الجميل ان نكون قد تعرفنا على كيفية جعل النافذة طبقية و لكن السؤال الحقيقي ... هو لماذا اجعل النافذة طبقية؟! .. جعل النافذة طبقية يمكنك تغيير خصائصها مثل جعل خلفية الفورم شفافة او على شكل صورة

طبعا توجد الكثير من الطرق لجعل الفورم تدعم الشفافية مثل ... الخاصية TransparentColor او العمل مع الـ Regions وما الى ذلك ... كما يمكن ايضا ان نحصل على نوع من الشفافية عن طريق الخاصية Opacity التابعة للفورم.

غير ان ما اتكلم عنه هو شيء مختلف لنفترض ان لدينا صورة جميلة من النوع PNG الداعم للشفافية و لنقل هذه الصورة هي صورة النحلة :



اذا كان ذوقك مقارب لذوقي فسوف تروق لك النحلة كما تروق لي بالفعل ... و لنفرض انني رغبت في ان اجعل الفورم تأخذ شكل هذه النحلة لتبدو مقاربة الى ... الشكل التالي :



اذا اردنا ان نقوم بذلك برمجيا ... فلربما فكرنا بعمل فورم وذلك بجعل الصورة خلفية لها في الخاصية BackgroundImage ونجعل لون الخلفية BackColor ابيض مثلا و نجعل الخاصية transparentColor ابيض ايضا .. وبهذا تصبح الاجزاء البيضاء شفافة كالتالي :

```
Private Sub Form1_Load(.....) Handles MyBase.Load
    Me.BackColor = Color.White
    Me.TransparentKey = Color.White
End Sub
```

و الناتج عن ذلك شيئا مشابها للتالي :

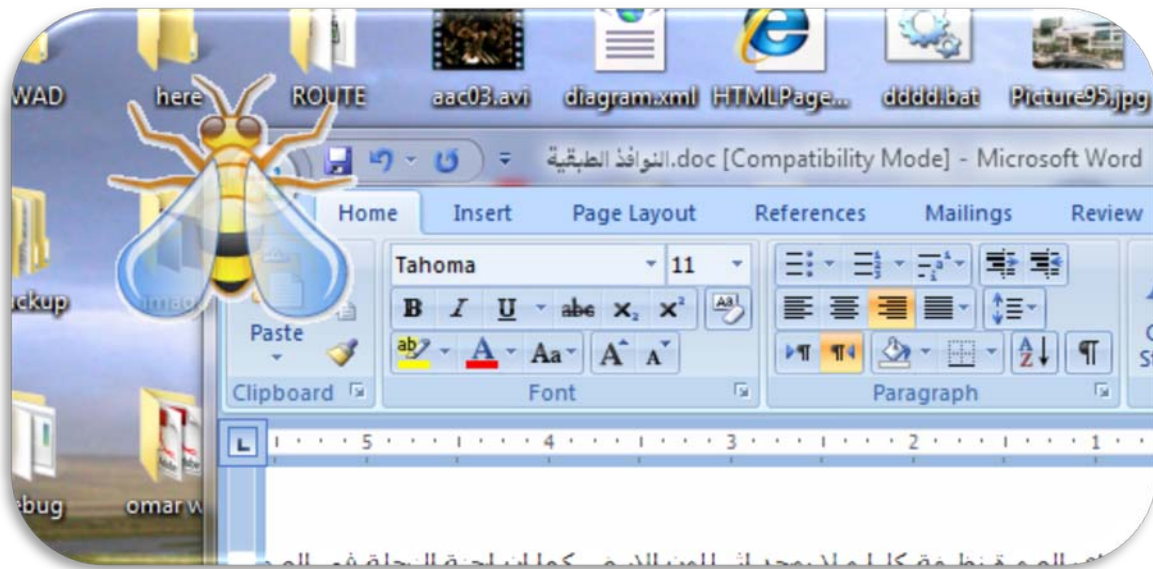


و بهذا نكون قد حققنا شفافية جميلة و بسطري فقط من اللود !! جميل غير انك اذا أمعنت النظر ستجد ان الاطراف يوجد عندها لو أبيض ... كما ان الاجزاء تحت الصورة (النحلة) غير ظاهرة جزئيا اما اذا رجت الى الصورة التي طلبناها فهي كالتالي :



ستلاحظ ان
اطراف الصورة
نظيفة كليا و لا
يوجد اثر للون
الابيض كما ان
اجنة النحلة
في الصورة
شفافة جزئيا
يمكنك رؤية ما
تحتها.

واذا اخترت ان تقوم بتعديل الخاصية Opacity التابعة للفورم فستجد ان الصورة كلها اصبحت شفافة كالتالي :



ستجد ان الصورة كلها اصبحت شفافة حتى الاجزاء التي ينبغي أن لا تكون شفافة وذلك لأن الخاصية Opacity لا تهتم بتفاصيل الصورة و إنما تهتم بشفافية الفورم بشكل عام.

و هنا نحتاج الى ان تكون اولا الفورم طبقية Layered Window و ثم تكون لدينا صورة ذات تدعم الشفافية Alpha Blending وفيها أجزاء شفافة و أخرى شبه شفافة و أخرى معتمة (و تجد هذا غالبا في الصور من النوع PNG)

لهذا الغرض كله قمت بتصميم فئة بسيطة تجعل بالامكان إستخدام بعضا من دوال الـ APIs لجعل الفورم شفافة وبحسب الصورة التي تريدها

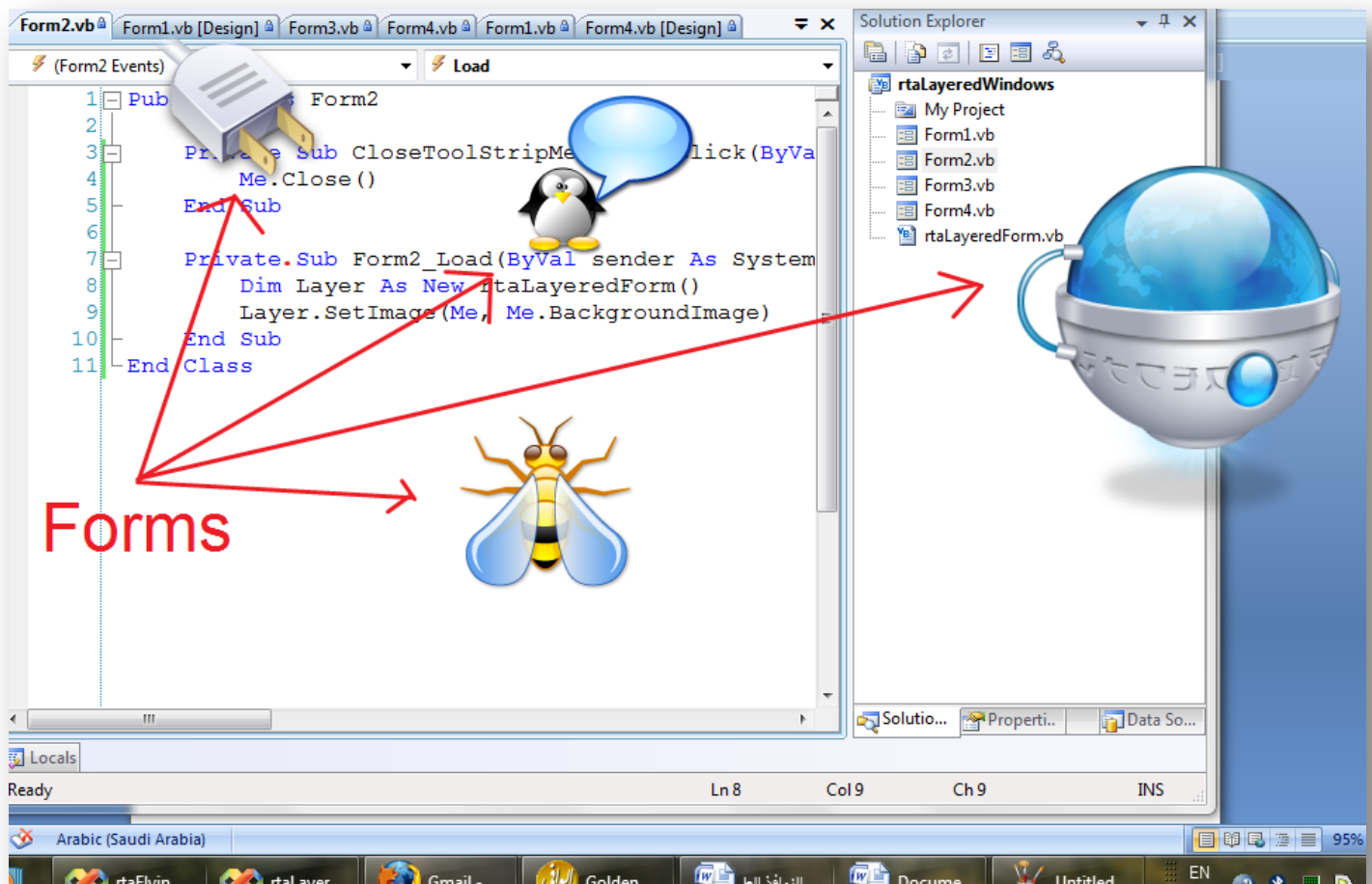
اسم الفئة التي صممتها هو `rtaLayeredForm` وهي عبارة عن فئة بسيطة تقوم بإستخدام بعض دوال الـ API للتحكم بمظهر الفورم ... لكي تستخدم الفئة وتجعل الفورم بشكل اي صورة تريدها ... ستحتاج الى كتابة شيء مثل هذا في الحدث Load للفورم :

```
Private Sub Form1_Load(.....) Handles MyBase.Load

    Dim Layer As New rtaLayeredForm()
    Layer.SetImage(Me, Me.BackgroundImage)

End Sub
```

امثلة على اشكال الفورم التي يمكن ان تنشئها :



إن ما تراه في الاعلى هنا عبارة عن فورمات في برنامجنا منتشرة هنا وهناك

```
Public Class rtaLayeredForm
```

```
Private mParent As Form
Public ReadOnly Property Parent() As Form
    Get
        Return mParent
    End Get
End Property
```

```
Private mMovable As Boolean = True
Public Property Movable() As String
    Get
        Return mMovable
    End Get
    Set(ByVal value As String)
        mMovable = value
    End Set
End Property
```

```
Public Sub SetImage(ByVal frm As Form, ByVal btm As Bitmap)
```

```
    Try
```

```
        mParent = frm
```

```
        AddHandler frm.MouseDown, AddressOf OnMouseDown
```

```
        AddHandler frm.MouseMove, AddressOf OnMouseMove
```

```
        Dim exStyle As Int32 = APIs.GetWindowLong(frm.Handle.ToInt32, APIs.GWL_EXSTYLE)
```

```
        APIs.SetWindowLong(frm.Handle.ToInt32, APIs.GWL_EXSTYLE, exStyle Or
```

```
        APIs.WS_EX_LAYERED)
```

```
        If Not (Bitmap.IsCanonicalPixelFormat(btm.PixelFormat)) Or Not
```

```
        (Bitmap.IsAlphaPixelFormat(btm.PixelFormat)) Then
```

```
            Throw New ApplicationException("The picture must be 32bit picture with alpha  
channel.")
```

```
        End If
```

```
        Dim hDC As Int32 = APIs.GetDC(0)
```

```
        Dim hBtm As Int32
```

```
        Dim memDC As Int32 = APIs.CreateCompatibleDC(hDC)
```

```
        Dim Bits As Int32
```

```
        Dim dstPos As New APIs.POINTAPI(frm.Left, frm.Top)
```

```
        Dim srcPos As New APIs.POINTAPI(0, 0)
```

```
        Dim Siz As New APIs.SizeAPI(btm.Width, btm.Height)
```

```
        Dim Opt As New APIs.BLEND_OPTIONS
```

```
        Opt.SourceConstantAlpha = 255
```

```
        Opt.AlphaFormat = 1
```

```
        Opt.BlendOp = 0
```

```
        Opt.BlendFlags = 0
```

```
        Try
```

```
            hBtm = btm.GetHbitmap(Color.Black).ToInt32
```

```
            Bits = APIs.SelectObject(memDC, hBtm)
```

```
            APIs.UpdateLayeredWindow(frm.Handle, hDC, dstPos, Siz, memDC, srcPos, 0, Opt, 2)
```

```
        Catch ex As Exception
```

```
    End Try
```

```
    If hBtm <> 0 Then
```

```
        APIs.SelectObject(memDC, Bits)
```

```
        APIs.DeleteObject(hBtm)
```

```
    End If
```

```
    APIs.ReleaseDC(0, hDC)
```

```

        APIs.DeleteDC(memDC)

    Catch ex As Exception

    End Try
End Sub

Private mLastLocation As Point
Private Sub OnMouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs)
    mLastLocation = e.Location
End Sub

Private Sub OnMouseMove(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs)
    If (mMovable) AndAlso (e.Button = MouseButtons.Left) Then
        Dim pos As Point = Parent.PointToScreen(e.Location)

        pos.Offset(-mLastLocation.X, -mLastLocation.Y)
        Parent.Location = pos
    End If
End Sub

End Class

```

وهذه الفئة بكل ال APIs التي استخدمتها ... وهي كالتالي :

```

Private Class APIs

    Public Declare Function CreateCompatibleDC Lib "gdi32.dll" (ByVal hdc As Int32) As Int32
    Public Declare Function GetDC Lib "user32.dll" (ByVal hwnd As Int32) As Int32
    Public Declare Function SelectObject Lib "gdi32.dll" (ByVal hdc As Int32, ByVal hObject
As Int32) As Int32
    Public Declare Function ReleaseDC Lib "user32.dll" (ByVal hwnd As Int32, ByVal hdc As
Int32) As Int32
    Public Declare Function DeleteDC Lib "gdi32.dll" (ByVal hdc As Int32) As Int32
    Public Declare Function DeleteObject Lib "gdi32.dll" (ByVal hObject As Int32) As Int32
    Public Declare Function UpdateLayeredWindow Lib "user32.dll" (ByVal hwnd As Int32, ByVal
hdcDst As Int32, ByRef pptDst As POINTAPI, ByRef psize As SizeAPI, ByVal hdcSrc As Int32, ByRef
pptSrc As POINTAPI, ByVal crKey As Int32, ByRef pblend As BLEND_OPTIONS, ByVal dwFlags As Int32)
As Int32
    Public Declare Function GetWindowLong Lib "user32.dll" Alias "GetWindowLongW" (ByVal hwnd
As Int32, ByVal nIndex As Int32) As Int32
    Public Declare Function SetWindowLong Lib "user32.dll" Alias "SetWindowLongW" (ByVal hwnd
As Int32, ByVal nIndex As Int32, ByVal dwNewLong As Int32) As Int32

    Public Const GWL_EXSTYLE As Int32 = -20
    Public Const WS_EX_LAYERED As Int32 = &H80000

    <StructLayout(LayoutKind.Sequential, Pack:=1)> _
    Public Structure BLEND_OPTIONS
        Public BlendOp As Byte
        Public BlendFlags As Byte
        Public SourceConstantAlpha As Byte
        Public AlphaFormat As Byte
    End Structure

    <StructLayout(LayoutKind.Sequential)> _
    Public Structure SizeAPI
        Public cx As Int32
        Public cy As Int32
    End Structure

```

```

        Sub New(ByVal cx As Int32, ByVal cy As Int32)
            Me.cx = cx
            Me.cy = cy
        End Sub
    End Structure

<StructLayout(LayoutKind.Sequential)> _
Public Structure POINTAPI
    Public x As Int32
    Public y As Int32

    Sub New(ByVal x As Int32, ByVal y As Int32)
        Me.x = x
        Me.y = y
    End Sub
End Structure

Public Declare Function ExtCreateRegion Lib "gdi32.dll" ( _
    ByRef lpXform As XFORM, _
    ByVal nCount As Int32, _
    ByRef lpRgnData As RGNDATA) As Int32
<StructLayout(LayoutKind.Sequential)> _
Public Structure XFORM
    Public eM11 As Single
    Public eM12 As Single
    Public eM21 As Single
    Public eM22 As Single
    Public eDx As Single
    Public eDy As Single
End Structure

<StructLayout(LayoutKind.Sequential)> _
Public Structure RGNDATA
    Public rdh As RGNDATAHEADER
    Public Buffer As Byte
End Structure

<StructLayout(LayoutKind.Sequential)> _
Public Structure RGNDATAHEADER
    Public dwSize As Int32
    Public iType As Int32
    Public nCount As Int32
    Public nRgnSize As Int32
    Public rcBound As RECT
End Structure

<StructLayout(LayoutKind.Sequential)> _
Public Structure RECT
    Public Left As Int32
    Public Top As Int32
    Public Right As Int32
    Public Bottom As Int32
End Structure

```

End Class

