



### اضافه کردن شناسنامه<sup>۱</sup> به ابزار ActiveX

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
95 / 98 / NT	5 , 6	۲۹ سپتامبر ۱۹۹۸	161008.TXT

ابزارهای ActiveX معمولاً یک خصوصیت About دارند که این امکان را به برنامه نویسان می دهد تا اطلاعات بیشتری در مورد سازنده ابزار مذکور مشاهده کنند. ویژوال بیسیک امکان ایجاد این خصوصیت About را برای ابزارهای ActiveX در اختیار می گذارد.

۱- یک پروژه ActiveX Control ایجاد کنید.

۲- در پنجره Project، روی ابزار ActiveX خود با دکمه راست کلیک کنید. از منوی ظاهر شده گزینه Add را انتخاب کرده و سپس گزینه Form را انتخاب کنید.

۳- در نگارشهای حرفه ای و فوق حرفه ای در لیست الگوهای فرم روی About Dialog دوبار کلیک کنید. در نگارش ویژه ایجاد ابزار<sup>۲</sup>، الگویی برای About Dialog وجود ندارد. در عوض از مراحل زیر پیروی کنید:

الف) در لیست الگوهای فرم روی Form دوبار کلیک کنید.

ب) نام فرم را به frmAbout تغییر دهید و عنوان پنجره را به About MyApp تغییر دهید. MyApp نام برنامه شماست.

پ) یک CommandButton به فرم اضافه کنید، عنوانش را OK اختیار کرده و خصوصیت Default آنرا به True تغییر دهید.

ت) کد زیر را به واقعه Click از ابزار CommandButton اضافه کنید:

```
Private Sub Command1_Click()
    Unload Me
End Sub
```

<sup>۱</sup> About Box

<sup>۲</sup> Control Creation Edition

ث) ابزارهای دیگر مورد نظر خود را به این فرم اضافه کنید تا ظاهری مطابق خواست شما داشته باشد. معمولاً در این موارد ابزارهایی که به کار می روند عبارتند از: Label هایی که نشان دهنده عنوان برنامه، شماره نگارش آن، توضیحات و از این قبیل باشند و یک PictureBox برای نمایش آیکون برنامه و نیز یک CommandButton برای نمایش اطلاعات سیستم<sup>۳</sup>.

۴- در پنجره Project، روی ابزار ActiveX خود با دکمه سمت راست کلیک نمایید. سپس گزینه View Code را انتخاب کرده و کد زیر را به ابزار ActiveX خود اضافه نمایید:

```
Sub ShowAboutBox()
    frmAbout.Show vbModal
    Unload frmAbout
    Set frmAbout = Nothing
End Sub
```

۵- از منوی Tools گزینه Procedure Attributes را انتخاب نمایید.

۶- روتین ShowAboutBox را در فیلد Name انتخاب کرده و دکمه >>Advanced را کلیک کنید.

۷- از فیلد Procedure ID آیتم AboutBox را انتخاب نموده و دکمه OK را کلیک کنید.

۸- تمام پنجره های پروژه ابزار ActiveX خود را ببندید. این کار ابزار را در حالت اجرا قرار می دهد.

۹- یک پروژه معمولی جدید ایجاد کنید (توسط گزینه Add Project در منوی File) و یکی از ابزارهای ActiveX خود را روی فرم آن قرار دهید. خصوصیت About در لیست خصوصیت های ابزار ظاهر می شود. اگر روی دکمه جلوی این خصوصیت کلیک کنید، frmAbout نمایش داده خواهد شد.

<sup>۳</sup> System Information

افزودن راهنمای حساس به وضعیت <sup>۴</sup> در ابزارهای ActiveX			
نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
180733.TXT	۲۴ آگوست ۱۹۹۸	5, 6	95 / 98 / NT

این بخش نشان می دهد چگونه یک راهنمای حساس به وضعیت، به یک ابزار ActiveX ساخته شده در ویژوال بیسیک اضافه نمایید. با افزودن این خاصیت به ابزار خود، به کاربران خود امکان می دهید سریعاً به اطلاعات و راهنماییهای مورد نیاز خود در مورد این ابزار دسترسی یابند. این در صورتی است که ابزار شما در یک فرم ویژوال بیسیک بکار رود. در صورتیکه این ابزار را در مرورگر وب Internet Explorer بکار ببرید، نتیجه دیگری خواهد داشت.

برای هر بخش از ابزار ActiveX خود یا ابزارهایی که در آن استفاده کرده اید می توانید یک سرفصل<sup>۵</sup> در فایل راهنما ایجاد نمایید. اولین مرحله اینست که یک فایل راهنما برای ابزار ActiveX خود تعیین کنید، که بعد از آن خواهید توانست برای خود ابزار ActiveX و ابزارهای بکار رفته در آن سرفصل هایی از راهنما را اختصاص دهید.

برای تعیین یک فایل راهنما برای یک ابزار ActiveX مراحل زیر را دنبال کنید:

- ۱- در پروژه ابزار ActiveX خود، از منوی Project Properties گزینه Project Properties را انتخاب کنید. پس از این کار پنجره مربوطه ظاهر می شود.
- ۲- در تابلو<sup>۶</sup> General نام فایل راهنمای مورد نظر خود را در مستطیل Help File Name وارد نمایید. این فایل راهنما باید حاوی تمامی موضوعات مورد استفاده در ابزار شما باشد.
- ۳- در مستطیل Project Help Context ID از همین تابلو، شماره مخصوص سرفصلی که مایلید برای ابزار ActiveX نمایش داده شود را وارد کنید.
- ۴- روی OK کلیک کنید تا پنجره Project Properties بسته شود.

<sup>۴</sup> Context-Sensitive Help

<sup>۵</sup> Topic

<sup>۶</sup> Tab

- به این ترتیب یک سرفصل از فایل راهنما را برای ابزار ActiveX خود تعریف کردید. پس از اینکه شما یکی از اعمال زیر را انجام دهید، این سرفصل از فایل راهنما ظاهر خواهد شد:
- در مرورگر اشیای ویژوال بیسیک، روی کلاس ابزار ActiveX خود با دکمه سمت راست کلیک کنید تا منوی مخصوص ظاهر شود. در این منو گزینه Help را انتخاب نمایید.
  - ابزار مذکور را روی یک فرم در یک پروژه معمولی قرار دهید و ابزار را انتخاب کرده و کلید F1 را بزنید.
  - هنگامی که پروژه ای حاوی ابزار مذکور اجرا شده باشد، تمرکز را روی ابزار منتقل کرده و کلید F1 را بزنید.
- علاوه بر این می توانید برای هر یک از ابزارهای بکار رفته در ابزار ActiveX خود یک سرفصل در فایل راهنما مشخص کنید. این کار توسط تنظیم خصوصیت HelpContextID هر ابزار صورت می گیرد. با این حال، این سرفصل های راهنما فقط زمانی نمایش داده می شوند که خصوصیات WhatsThisButton و WhatsThisHelp مربوط به پنجره ای که ابزار شما روی آن قرار دارد، مقدار True داشته باشند. فقط زمانی که کاربر روی دکمه WhatsThisButton کلیک کند و علامت سؤال را روی ابزار خاصی قرار دهد، راهنمای مربوط به آن ابزار نمایش داده می شود.
- اگر تمرکز برنامه روی ابزار مورد استفاده در ابزار اصلی شما باشد و کاربر کلید F1 را فشار دهد، سرفصلی از راهنما که برای ابزار اصلی تعیین شده است به جای سرفصل مخصوص همان ابزار فرعی، نمایش داده می شود.

تنظیم اندازه حداقل و حداکثر برای یک ابزار ActiveX			
نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
190224.TXT	۲۲ اکتبر ۱۹۹۸	5, 6	95 / 98 / NT

هنگامی که در ویژوال بیسیک ابزار می سازید، شاید بخواهید ابعاد ابزار خود را محدود کنید. در این بخش خواهید دید که چگونه ابعاد حداقل و حداکثر برای ابزار دست ساز خود تعریف کنید تا برنامه سازانی که از ابزار شما استفاده می کنند نتوانند ابعادی فراتر از این حداقل و حداکثر به ابزار شما بدهند.

این مثال از یک ابزار دست ساز در یک پروژه معمولی استفاده می کند. شما می توانید همین تکنیک را برای ابزار دست ساز در یک پروژه ابزار ActiveX بکار ببرید. هنگامی که ابزار ActiveX ساخته شد، هر کس که از آن استفاده می کند نخواهد توانست ابعاد آنرا بزرگتر یا کوچکتر از محدوده ای که شما در خود ابزار تعریف کرده اید تنظیم کند.

توجه: این روش فقط در برنامه های ویژوال بیسیک عمل خواهد کرد و در دیگر محیط ها مانند مایکروسافت Word یا Excel ممکنست عمل نکند.

- ۱- یک پروژه معمولی جدید ایجاد کنید. Form1 بطور پیش فرض بوجود می آید.
- ۲- از منوی Project گزینه Add UserControl را انتخاب کنید تا یک ابزار دست ساز به پروژه اضافه شود.
- ۳- خصوصیت BackColor از UserControl را به رنگی متفاوت مانند قرمز تغییر دهید.
- ۴- دو Label به UserControl اضافه کنید.
- ۵- کد زیر را به پنجره کد UserControl اضافه کنید:

```
Private Sub UserControl_Resize()
' Check to see if the control is larger or smaller than the preset
' minimum or maximum size.
' If it is larger or smaller, set the size of the control.
Select Case Height
Case Is < 2400
Height = 2400
Case Is > 3600
Height = 3600
End Select
```

```

Select Case Width
Case Is < 2400
Width = 2400
Case Is > 3600
Width = 3600
End Select
' The label reports the height and width of the control in
' twips even if the container has a different scalemode.
Label1.Caption = "Height: " & Height
Label2.Caption = "Width: " & Width
End Sub

```

۶- UserControl را ببندید تا در جعبه ابزار<sup>۷</sup> ظاهر شود. آنرا روی Form1 قرار دهید.

۷- سعی کنید ابعاد ابزار دست ساز خود را تغییر دهید. اگر سعی کنید ابعاد ابزار را بزرگتر یا کوچکتر از محدوده تعریف شده تنظیم کنید، ابعاد ابزار به حداکثر یا حداقل بازخواهد گشت.

### ایجاد ابزارهای مرتبط با داده های بانک های اطلاعاتی

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
161273.TXT	۱۶ جون ۱۹۹۹	5, 6	95 / 98 / NT

در ویژوال بیسیک این امکان وجود دارد که برنامه نویسی ابزارهای دست ساز خود را به نحوی طراحی کند که خصوصیت‌هایی مرتبط با داده های یک بانک اطلاعاتی داشته باشد. روش تعریف چنین خصوصیت‌هایی را می‌توان به طور خلاصه طی مراحل زیر بیان کرد:

۱- در حالیکه پنجره نشان دهنده لیست بیسیک ابزار ActiveX قابل مشاهده است، از منوی Tools گزینه Procedure Attributes را انتخاب نمایید.

۲- در پنجره ای که ظاهر می‌شود دکمه Advanced را فشار دهید.

<sup>۷</sup> ToolBox

۳- در قسمت Name ، خصوصیت مورد نظر خود را انتخاب کنید.

برای خصوصیت های قابل ارتباط با داده ها، گزینه های زیر را باید بر حسب مورد استفاده نمود:

الف - Property binds to DataField

مشخص کردن این گزینه به این معنا است که خصوصیت DataField، که در زمان طراحی ( design-time) قابل استفاده است، فیلدی از بانک اطلاعاتی را مشخص می کند که خصوصیت مورد نظر به آن مرتبط می شود. فقط یک خصوصیت می تواند از خصوصیت DataField برای ارتباط با فیلدی از بانک اطلاعاتی استفاده نماید. اگر بیش از یک خصوصیت مرتبط با بانک اطلاعاتی در ابزار دست ساز خود دارید، باید از مجموعه DataBindings (که در بخش بعد شرح داده شده) استفاده کنید.

ب - Show in DataBindings collection at design-time

استفاده از این گزینه باعث می شود خصوصیت DataBindings در زمان طراحی قابل استفاده باشد. هنگامی که از این گزینه استفاده شود، کلیک کردن روی این خصوصیت (در پنجره خصوصیت ها) پنجره ای نمایش خواهد داد که در آن می توان مشخص نمود که کدام خصوصیت از ابزار ما با کدام فیلد از بانک داده ارتباط دارد.

پ - Property will call CanPropertyChange before changing

به کمک این گزینه می توان توسط تابع CanPropertyChange اطمینان حاصل کرد که آیا امکان تغییر مقدار خصوصیت مرتبط با داده ها (که باعث تغییر داده می شود) وجود دارد یا خیر.

نحوه ساخت یک ابزار ActiveX دست ساز که دارای دو خصوصیت مرتبط با داده ها است در مثال زیر نشان داده شده است:

۱- یک پروژه جدید از نوع ActiveX ایجاد کنید. UserControl1 بطور پیش فرض بوجود می آید.

۲- دو ابزار Textbox به نامهای Text1 و Text2 به UserControl1 اضافه کنید.

۳- کد زیر را به بخش Declarations از UserControl1 اضافه نمایید:



```

Option Explicit
Private Sub UserControl_Initialize()
    Text1.Text = "Caption"
    Text1.Left = 0
    Text2.Text = "Subcaption"
    Text2.Left = 0
    Text2.Font.Name = "Arial"
    Text2.Font.Size = 7
End Sub

Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    Caption = PropBag.ReadProperty("Caption", "")
    SubCaption = PropBag.ReadProperty("SubCaption", "")
End Sub

Property Let Caption(strNewValue As String)
    If CanPropertyChange("Caption") Then
        Text1.Text = strNewValue
        PropertyChanged "Caption"
    End If
End Property

Property Get Caption() As String
    Caption = Text1.Text
End Property

Property Let SubCaption(strNewValue As String)
    If CanPropertyChange("SubCaption") Then
        Text2.Text = strNewValue
        PropertyChanged "Caption"
    End If
End Property

Property Get SubCaption() As String
    SubCaption = Text2.Text
End Property

Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
    PropBag.WriteProperty "Caption", Caption, ""
    PropBag.WriteProperty "SubCaption", SubCaption, ""
End Sub

Private Sub Text1_Change()
    PropertyChanged "Caption"
End Sub

Private Sub Text2_Change()
    PropertyChanged "SubCaption"
End Sub

```

۴- برای مرتبط کردن خصوصیت های Caption و SubCaption به داده های بانک اطلاعاتی، از

منوی Tools گزینه Procedure Attributes را انتخاب کنید.

۵- در بخش Name خصوصیت Caption را انتخاب کنید و روی دکمه Advanced کلیک کنید.

- ۶- گزینه Property is data bound را انتخاب نمایید.
- ۷- گزینه Show in DataBindings collection at design-time را انتخاب کنید.
- ۸- گزینه Property will call CanPropertyChange before changing را انتخاب کنید.
- ۹- گزینه This property binds to DataField را انتخاب کنید.
- ۱۰- روی دکمه Apply کلیک نمایید.
- ۱۱- مراحل ۶ تا ۸ را برای خصوصیت SubCaption تکرار کنید و روی دکمه Apply کلیک نمایید. مرحله ۹ را نباید برای این خصوصیت استفاده کنید زیرا فقط یک خصوصیت می تواند به خصوصیت DataField مرتبط شود.
- ۱۲- روی دکمه OK کلیک کنید.
- اکنون خصوصیت Caption بگونه ای تعریف شده است که می تواند از طریق خصوصیت DataField به یک فیلد از داده های بانک داده مرتبط شود. خصوصیت SubCaption بگونه ای تعریف شده است که می تواند از طریق مجموعه DataBindings به فیلدی از بانک اطلاعاتی پیوند یابد.

### ساخت ابزارهای سبک وزن توسط ویژوال بیسیک ۶

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
184645.TXT	۱۵ جولای ۱۹۹۹	6	95 / 98 / NT

ابزارهای سبک وزن یا بدون پنجره، ابزارهایی هستند مشابه ابزارهای معمولی با این تفاوت که ابزارهای سبک وزن دارای شناسه پنجره نیستند. نداشتن شناسه پنجره باعث می شود که این ابزارها به منابع سیستم کمتری نیاز داشته باشند و به این ترتیب برای کار در برنامه های

اینترنت، برنامه های گسترده<sup>8</sup> و برنامه هایی که نیازمند این خصوصیت هستند، ایده آل باشند. به علاوه، ابزارهای سبک وزن می توانند اشکال دلخواه داشته و نیز دارای خصوصیت های ویژه ای باشند.

هنگامی که یک ابزار سبک وزن ایجاد می کنید، فقط می توانید در ساخت آن از ابزارهای سبک وزن دیگر استفاده نمایید. اگر سعی کنید یک ابزار معمولی روی این ابزار بکار ببرید، پیغام خطای زیر را مشاهده خواهید نمود:

```
Can't have windowed child controls on a windowless control.
```

ابزارهای معمولی همیشه در بالای ابزارهای سبک وزن قرار می گیرند زیرا ابزارهای سبک وزن از منابع سیستم مربوط به فرم مادر استفاده می کنند. خصوصیت ZOrder یک ابزار معمولی فقط نسبت به ابزارهای معمولی دیگر تغییر می کند.

از آنجا که ابزارهای سبک وزن دارای خصوصیت Hwnd نیستند، نمی توانید آنها را با توابع API که نیازمند پارامتری به نام Hwnd هستند به کار ببرید. در عوض، خصوصیت Hwnd محیط در برگیرنده ابزار، مورد استفاده قرار می گیرد که ممکنست باعث ایجاد نتایج غیر قابل پیش بینی گردد.

برای تبدیل یک ابزار معمولی به یک ابزار سبک وزن کفایت خصوصیت Windowless آنرا به مقدار True تغییر دهید.

ویژوال بیسیک به همراه یکسری ابزارهای سبک وزن ارائه شده است که در فایل MSWLess.ocx قرار دارند. برای کسب اطلاعات بیشتر در مورد این ابزارها و نحوه استفاده از آنها به بخش شماره ۱۸۴۶۸۷ از بانک اطلاعات مایکروسافت رجوع کنید:

Q184687 INFO: Lightweight Controls in Visual Basic 6.0

<sup>8</sup> Distributed Applications

در مثال زیر نشان داده می شود که چگونه یک ابزار سبک وزن ایجاد می شود. همچنین این مثال تفاوت بین ابزارهای سبک وزن و معمولی را از بابت میزان منابع سیستم مورد استفاده نشان می دهد.

در این مثال، مدت زمان لازم برای بارکردن ابزار سبک وزن به حافظه و خارج کردن آن از حافظه با یک ابزار معمولی مقایسه شده است. برای استفاده از این مثال لازم است یک پروژه EXE، یک ابزار سبک وزن و یک ابزار معمولی ساخته شود. مراحل ساخت برنامه EXE:

۱- یک پروژه EXE جدید در ویژوال بیسیک ایجاد کنید. Form1 بطور پیش فرض ایجاد می شود.

۲- ابزارهای زیر را به Form1 اضافه کنید و خصوصیت های آنها را مطابق جدول تنظیم کنید:

ابزار	نام پیش فرض	خصوصیت	مقدار خصوصیت
Command Button	Command1	Name Height Left Top Width	cmdWndLess 375 120 840 2535
Command Button	Command2	Name Height Left Top Width	cmdWnd 375 3000 840 2535
TextBox	Text1	Name Height Left Top Width	txtNumofControls 285 1680 120 375
Label	Label1	Name AutoSize Height Left Top	lblWndLessLoad True 195 120 1320
Label	Label2	Name AutoSize Height Left Top	lblWndLessUnload True 195 120 1680

Label	Label3	Name	lblWndLoad
		AutoSize	True
		Height	195
		Left	3000
		Top	1320
Label	Label4	Name	lblWndUnload
		AutoSize	True
		Height	195
		Left	3000
		Top	1680
		Width	480
Label	Label5	Name	lblStartTicks
		Alignment	Right Justify
		AutoSize	True
		Caption	Start Ticker
		Height	195
		Left	4680
		Top	480
		Width	810
Label	Label6	Name	lblEndTicks
		Alignment	Right Justify
		AutoSize	True
		Caption	End Ticks:
		Height	195
		Left	4680
		Top	120

۳- کد زیر را در Form1 وارد کنید:

```
Option Explicit
Private Declare Function GetTickCount Lib "kernel32" () As Long
Dim bLoadWnd As Boolean
Dim bLoadWndLess As Boolean
Dim lTickStart As Long
Dim lTickEnd As Long
Dim lResult As Long
Dim colCtrlWnd As Collection
Dim colCtrlWndLess As Collection

Private Sub cmdWnd_Click()
'This event loads and unloads regular controls.
Dim tmpCtrlWnd As Control
Dim lCounter As Long

If bLoadWnd = True Then
'Load Regular controls.
lblWndLoad.Caption = "Loading Regular Controls"
lTickStart = GetTickCount
lblStartTicks.Caption = "Start: " & CStr(lTickStart) & " ms"

For lCounter = 1 To Val(txtNumofControls.Text)
Set tmpCtrlWnd = Controls.Add("TestControls.ctrlWnd", _
"ctrlWnd" & lCounter)
colCtrlWnd.Add tmpCtrlWnd
```

```

        Set tmpCtrlWnd = Nothing
        DoEvents
    Next

    lTickEnd = GetTickCount
    lblEndTicks.Caption = "End: " & CStr(lTickEnd) & " ms"
    lblWndLoad.Caption = "Load Time: " & _
        CStr(lTickEnd - lTickStart) & " ms"

    bLoadWnd = False
    cmdWnd.Caption = "Unload Regular Controls"

Else
    'Unload regular controls.
    lblWndUnload.Caption = "Unloading Regular Controls"
    lTickStart = GetTickCount
    lblStartTicks.Caption = "Start: " & CStr(lTickStart) & " ms"
    ' Unload controls.
    For lCounter = colCtrlWnd.Count To 1 Step -1
        Controls.Remove colCtrlWnd.Item(lCounter).Name
        colCtrlWnd.Remove (lCounter)
    Next
    lTickEnd = GetTickCount
    lblEndTicks.Caption = "End: " & CStr(lTickEnd) & " ms"
    lblWndUnload.Caption = "Unload Time: " & _
        CStr(lTickEnd - lTickStart) & " ms"

    bLoadWnd = True
    cmdWnd.Caption = "Load Regular Controls"
End If

End Sub

Private Sub cmdWndless_Click()
    'This command button loads and unloads the lightweight controls.

    Dim tmpCtrlWndLess As Control
    Dim lCounter As Long
    If bLoadWndLess = True Then
        lblWndLessLoad.Caption = "Loading Lightweight Controls"
        lTickStart = GetTickCount
        lblStartTicks.Caption = "Start: " & CStr(lTickStart) & " ms"
        ' Load controls.
        For lCounter = 1 To Val(txtNumofControls.Text)
            Set tmpCtrlWndLess = _
                Controls.Add("TestControls.ctrlWndLess", _
                    "ctrlWndLess" & lCounter)
            colCtrlWndLess.Add tmpCtrlWndLess
            Set tmpCtrlWndLess = Nothing
            DoEvents
        Next

        lTickEnd = GetTickCount
        lblEndTicks.Caption = "End: " & CStr(lTickEnd) & " ms"
        lblWndLessLoad.Caption = "Load Time: " & _
            CStr(lTickEnd - lTickStart) & " ms"

        bLoadWndLess = False
        cmdWndless.Caption = "Unload Windowless Controls"
    Else
        lblWndLessUnload.Caption = "Unloading Lightweight Controls"
        lTickStart = GetTickCount
    
```

```

lblStartTicks.Caption = "Start: " & CStr(lTickStart) & " ms"
' UnLoad controls.
For lCounter = colCtrlWndLess.Count To 1 Step -1
    Controls.Remove colCtrlWndLess.Item(lCounter).Name
    colCtrlWndLess.Remove (lCounter)
Next

lTickEnd = GetTickCount
lblEndTicks.Caption = "End: " & CStr(lTickEnd) & " ms"
lblWndLessUnload.Caption = "Unload Time: " & _
    CStr(lTickEnd - lTickStart) & " ms"

bLoadWndLess = True
cmdWndless.Caption = "Load Windowless Controls"
End If
End Sub

Private Sub Form_Load()
    Set colCtrlWnd = New Collection
    Set colCtrlWndLess = New Collection
    cmdWndless.Caption = "Load Lightweight Controls"
    bLoadWndLess = True
    cmdWnd.Caption = "Load Regular Controls"
    bLoadWnd = True
    txtNumofControls.Text = "20"
End Sub

```

۴- از منوی Run گزینه Start را انتخاب کنید یا F5 را فشار دهید تا برنامه اجرا شود. اگر فرم به درستی نمایش داده شد نشانه اینست که عمل ایجاد برنامه EXE با موفقیت انجام شده است.

#### مراحل ساخت ابزارهای معمولی و سبک وزن:

- ۱- یک پروژه ابزار ActiveX به گروه پروژه های خود اضافه کنید.
  - از منوی File گزینه Add Project را انتخاب کنید. پنجره Add Project ظاهر می شود.
  - از بخش New ، گزینه ActiveX را انتخاب کنید.
  - روی دکمه Open کلیک کنید تا پروژه جدید به گروه پروژه های شما اضافه شود و پنجره Add Project بسته شود. به این ترتیب UserControl1 به طور پیش فرض ایجاد می شود.
  - نام UserControl1 را به ctrlWnd تغییر دهید. این یکی همان ابزار معمولی خواهد بود.
- ۲- یک Combo Box به ابزار ctrlWnd اضافه کنید. ابزار را ذخیره کنید.
- ۳- به نحو گفته شده در زیر یک UserControl دیگر به پروژه اضافه کنید. بدین ترتیب ابزار سبک وزن بوجود خواهد آمد:

- از منوی Project گزینه Add User Control را انتخاب کنید. پنجره Add User Control ظاهر می شود.
  - روی دکمه Open کلیک کنید تا پروژه جدید به گروه پروژه های شما اضافه شود و پنجره Add Project بسته شود. به این ترتیب UserControl1 به طور پیش فرض ایجاد می شود.
  - نام UserControl1 را به ctrlWndLess تغییر دهید.
  - به خصوصیت Windowless مقدار True بدهید.
  - ابزار را ذخیره نمایید.
  - نام پروژه را به TestControls تغییر دهید و پروژه را ذخیره نمایید.
- به این ترتیب مراحل ساخت ابزارهای ActiveX مورد نیاز برای این مثال به پایان می رسد. برای اجرای برنامه کلید F5 را فشار دهید یا از منوی Run گزینه Start را انتخاب نمایید. می توانید تعداد ابزارها را تغییر دهید و زمان بازیابی شدن برنامه را مقایسه نمایید.

### ساخت ابزار مرتبط با اینترنت

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
95 / 98 / NT	5 , 6	۱۰ جون ۱۹۹۹	185539.TXT

در این بخش روش ایجاد یک ابزار مرتبط با اینترنت نشان داده می شود که تصاویری را از یک سایت عمومی مستقر در اینترنت دریافت می کند.

۱- یک ابزار ActiveX جدید در ویژوال بیسیک ایجاد کنید. این کار باعث می شود UserControl بوجود آید. یک Picture Control و یک Label روی این UserControl قرار دهید به نحوی که Label در بالای Picture Control قرار داشته باشد. کد زیر را به UserControl اضافه کنید:

```
Option Explicit
Const vbAsyncTypePicture = 0
```



```

Private Sub UserControl_AsyncReadComplete(AsyncProp As AsyncProperty)
    On Error Resume Next
    Set Picture1.Picture = AsyncProp.Value
End Sub

Private Sub UserControl_Initialize()
    'PLEASE NOTE:
    'The URL could be in the form of a shared directory,
    'FTP or a HTTP. The convention below, is used as
    'an example only. Do not try to use this URL as it is
    'in fact bogus.
    UserControl.AsyncRead "http://example.microsoft.com/example.jpg", _
        vbAsyncTypePicture
End Sub

```

۲- خصوصیت Caption مربوط به Label را به شکل زیر تغییر دهید:

```
Label1.Caption = "Image View"
```

۳- نام پروژه را به چیزی غیر از Project1 تغییر دهید و ابزار را کامپایل نمایید تا فایل OCX بوجود آید.

۴- یک پروژه EXE جدید ایجاد کنید و ابزار تازه ساخته شده را به آن اضافه کنید. هنگامی که ابزار را روی فرم قرار می دهید، تصویر مورد نظر بطور خودکار در آن ظاهر می شود. (البته برای این آزمایش باید به اینترنت متصل باشید.)

می توان به سادگی یک Timer به این ابزار اضافه کرد و از آن برای تغییر دادن شماره تصویر استفاده نمود. همچنین می توان یک متغیر عمومی در این ابزار تعریف کرد تا بتوان به کمک آن از بیرون تصویر را تغییر داد.