

حذف آیتمها از منوی کنترل یک پنجره

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
3.1	3	۲۱ می ۱۹۹۸	110393.TXT

منوی کنترل در سمت چپ بالای هر پنجره ویژوال بیسیک قرار دارد. آیتم های مشخصی از این منو را می توان با تنظیم خصوصیات `MaxButton` ، `MinButton` و `BorderStyle` همان پنجره، از این منو حذف کرد. روش دیگر این کار استفاده از توابع API است که در مثال زیر خواهید دید. برای حذف کامل این منو می توان خصوصیت `ControlBox` پنجره را مساوی مقدار `False` قرار داد.

منوی کنترل بطور پیش فرض حاوی گزینه های زیر است:

```
Restore
Move
Size
Minimize
Maximize
-----
Close           Alt+F4
-----
Switch to...   Ctrl+Esc
```

می توانید با تنظیم خصوصیات `MaxButton` ، `MinButton` و `BorderStyle` پنجره، آیتم های متناظر را از منوی فوق حذف نمایید.

خصوصیت MaxButton :

دادن مقدار `False` به این خصوصیت در هنگام طراحی برنامه باعث می شود گزینه `Maximize` از منوی کنترل حذف شده و علامت ماکزیمایز در سمت راست پنجره نیز ناپدید شود. به علاوه این کار باعث می شود که دوبار کلیک کردن روی نوار عنوان پنجره موجب ماکزیمایز شدن پنجره نشود.

خصوصیت MinButton :

دادن مقدار False به این خصوصیت در هنگام طراحی برنامه باعث می شود گزینه Minimize از منوی کنترل حذف شده و علامت مینیمایز در سمت راست پنجره نیز ناپدید شود. به علاوه این کار باعث می شود که دوبار کلیک کردن روی نوار عنوان پنجره موجب ماکزیمایز شدن پنجره نشود.

خصوصیت BorderStyle :

جدول زیر نشان دهنده هر یک از مقادیر این خصوصیت به همراه نتیجه آن می باشد:

مقدار	شرح
0	پنجره بدون کادر و بدون عوامل مربوط به کادر خواهد بود.
1	کادر ثابت تک خطی - می تواند دارای منوی کنترل، نوار عنوان، دکمه ماکزیمایز و مینیمایز باشد. فقط توسط دکمه های مذکور می توان آنرا تغییر اندازه داد.
2	(پیش فرض) قابل تغییر اندازه - و دارای تمام امکانات گزینه ۱
3	کادر ثابت دوخطی - می تواند دارای منوی کنترل و نوار عنوان باشد. قابل تغییر اندازه نبوده و نمی تواند دارای دکمه های ماکزیمایز و مینیمایز باشد.

مثال زیر برای حذف تمام آیتم های منوی کنترل غیر از Minimize و Restore ، از فراخوانی

توابع API استفاده می کند:

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

توجه: در این مثال باید خصوصیات زیر بدون تغییر با مقادیر پیش فرض خود باقی

بمانند:

ControlBox = True MaxButton = True MinButton = True

توابع API تغییرات لازم را در این خصوصیات بوجود خواهند آورد.

۲- کد زیر را به واقعه Form_Load اضافه کنید:

```
Sub Form_Load ()
    Dim hSysMenu%, r%, j%, dw&, rr&
    Const MF_BYPOSITION = &H400
    ' Me refers to the form where code is currently executing:
    hSysMenu = GetSystemMenu(Me.hWnd, 0)
    For j = 8 To 4 Step -1
        r = RemoveMenu(hSysMenu, j, MF_BYPOSITION)
    Next j
    For j = 2 To 1 Step -1
        r = RemoveMenu(hSysMenu, j, MF_BYPOSITION)
    Next j
    ' Leave the Restore and Minimize items.
    dw& = GetWindowLong(Me.hWnd, -16) 'Window style
    dw& = dw& And &HFFFFFF 'Turn off bits for Maximize arrow button
    rr& = SetWindowLong(Me.hWnd, -16, dw&)
End Sub
```

آیتم های پیش فرض منوی کنترل از بالا به پایین دارای شماره های 0 تا 8 هستند. تمام یا بعضی از این آیتم ها را می توان به کمک توابع API حذف نمود. توجه کنید که باید این آیتم ها به ترتیب معکوس حذف گردند، از 8 تا 0، در غیر اینصورت شماره گذاری آنها به هم خواهد خورد.

۳- یک CommandButton به فرم خود اضافه کنید و کد زیر را در واقعه کلیک آن قرار دهید:

```
Sub Command1_Click ()
    End
End Sub
```

از آنجا که گزینه Close از منوی کنترل حذف خواهد شد، این دکمه امکان خروج از برنامه را فراهم می آورد.

۴- کد زیر را به بخش Declarations اضافه کنید:

```
' Enter each of the following Declare statements as one, single line:
Declare Function RemoveMenu% Lib "User" (ByVal hMenu%, ByVal nPosition%,
    ByVal wFlags%)
Declare Function GetSystemMenu% Lib "User" (ByVal hWnd%, ByVal revert%)
Declare Function GetWindowLong Lib "User" (ByVal hWnd As Integer,
    ByVal nIndex As Integer) As Long
Declare Function SetWindowLong Lib "User" (ByVal hWnd As Integer,
    ByVal nIndex As Integer, ByVal dwNewLong As Long) As Long
```

۵- برنامه را اجرا کنید.

منوی کنترل این پنجره دارای آیتم های Restore (بصورت خاکستری غیر فعال) و Minimize خواهد بود.

ایجاد یک پنجره بدون نوار عنوان

برای اینکه پنجره ای بدون نوار عنوان در ویژوال بیسیک ایجاد کنید که دارای کادر باشد، باید خصوصیت Caption پنجره را معادل تهی ("") تنظیم کنید، به BorderStyle مقدار ۱ (کادر ثابت تک خطی) یا ۲ (کادر قابل تغییر اندازه) یا ۳ (کادر ثابت دوخطی) داده و ControlBox، MaxButton و MinButton هم باید Fasle باشند.

اگر هر متنی (حتی یک کاراکتر فاصله) در خصوصیت Caption وجود داشته باشد یا یکی از خصوصیات ControlBox، MaxButton و MinButton مقدار True داشته باشند، نوار عنوان روی فرم ظاهر خواهد شد. توجه داشته باشید که قرار دادن مقدار صفر در BorderStyle موجب می شود که پنجره نوار عنوان نداشته باشد.

افزودن آیتم به منوی کنترل یک پنجره

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
110498.TXT	۱۲ می ۱۹۹۸	3	3.1

برای اضافه کردن آیتم به منوی کنترل یک پنجره در ویژوال بیسیک می توانید از تابع AppendMenu استفاده نمایید. با این حال ویژوال بیسیک قادر نخواهد بود به طور مستقیم وقایع مربوط به آیتم های جدید را تشخیص دهد. برای دریافت پیامهای مربوط به آیتم جدید می توانید از یک ابزار کنترل پیام استفاده کنید. چنین ابزاری را می توان توسط زبان C ایجاد نمود، اما ساخت آن توسط ویژوال بیسیک عملی نیست. روش دیگر اینست که از ابزارهای آماده مانند SpyWorks تولید شرکت Desaware استفاده کنید.

در برنامه نویسی ویندوز اصطلاح کنترل پیام به معنی مجموعه عملیاتی است شامل ایجاد روتین های کنترل پیام برای دریافت پیامهای یک پنجره بخصوص و سپس استفاده از پیامهای مورد نظر و در نهایت ارسال بقیه پیامها به کنترل کننده اصلی پیامهای آن پنجره. روتین مذکور فیلتری برای پیامهاست که برای بعضی از پیامها پردازشهایی انجام میدهد و بقیه پیامها را توسط تابع CallWindowProc به روتین اصلی کنترل پیام پنجره ارسال می نماید. تابع CallWindowProc پیام را برای سیستم ویندوز ارسال کرده که این سیستم بنوبه خود پیام را برای پنجره مربوطه خواهد فرستاد.

چگونه با Desaware تماس بگیریم

Desaware
5 Town & Country Village #790
San Jose, CA 95128
Contact: Dan Appleman (408) 377-4770
Fax: (408) 371-3530

توجه :

- ۱- محصولات این شرکت بدون هیچگونه ارتباطی با مایکروسافت تولید شده اند.
- ۲- اطلاعات فوق ممکنست تغییر نماید.

تذکر مترجم :

این بخش مخصوص نگارش ۳ ویژوال بیسیک نوشته شده است. امکان انجام عملیات کنترل پیام در نگارش های بالاتر ویژوال بیسیک میسر شده است. برای اطلاعات بیشتر به بخش شماره 185451 از همین فصل مراجعه نمایید.

افزودن منو به یک فرم از طریق Add-in			
نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
137038.TXT	۳۰ اکتبر ۱۹۹۵	4	All

در این بخش خواهیم دید که چگونه از داخل یک Add-in و با کمک فرمان AddMenuTemplate منویی به یک فرم اضافه کنیم. شیء FormTemplate که در مجموعه اشیای محیط برنامه سازی ویژوال بیسیک قرار دارد، دارای فرمانی است به نام AddMenuTemplate. این فرمان این امکان را فراهم می آورد که یک Add-in منوهای به یک فرم اضافه کند. این فرمان توان اضافه کردن منوهای اصلی و زیر منوها را دارد.

نحوه استفاده از فرمان AddMenuTemplate بصورت زیر است:

```
Menu = Form.AddMenuTemplate(Name, Parent)
```

که در آن :

Form : شیئی از نوع FormTemplate

Name : رشته ای که مشخص کننده خصوصیت Name برای منوی جدید است.

Parent : شیئی از نوع ControlTemplate که مشخص کننده منویی است که منوی جدید در داخل آن (زیر منوی آن) خواهد بود.

Menu : شیئی از نوع ControlTemplate که خود منوی جدید خواهد بود.

برای اینکه منوی اصلی جدیدی به یک فرم اضافه کنید، باید Parent را معادل Nothing

تعریف کنید. سطر زیر روش انجام این کار را نشان می دهد:

```
Set MyMenu = MyForm.AddMenuTemplate("mnuFile", Nothing)
```

برای اینکه یک زیر منو اضافه کنید، باید یک منوی موجود را به عنوان Parent مشخص

کنید. بصورت زیر:

```
Set MySubMenu = MyForm.AddMenuTemplate("mnuFileOpen", MyMenu)
```

فرمان AddMenuTemplate منوهای جدید را به انتهای منوهای موجود اضافه می کند. منوها را نمی توان بین دو منوی موجود وارد کرد.

پس از اینکه یک منوی جدید اضافه کردید، می توانید خصوصیاتش را تغییر دهید تا:

- ◆ عنوانش تغییر کند
- ◆ کلید میانبر به آن اضافه کنید
- ◆ یک جداکننده در منو ایجاد کنید
- ◆ فعالش کنید
- ◆ آنرا مخفی کنید
- ◆ آنرا به آرایه ای از کنترل ها تبدیل کنید
- ◆ و کارهای دیگری روی آن انجام دهید.

شیی که توسط فرمان AddMenuTemplate برگردانده می شود شییی از نوع ControlTemplate است که مشخص کننده منوی جدید است. خصوصیت های منو را می توان از طریق مجموعه^۱ خصوصیت های این ControlTemplate دستکاری نمود. برای مثال، برای تغییر عنوان منو از این روش استفاده کنید:

```
MyMenu.Properties("&Caption").Value = "New Caption"
```

برای اضافه کردن کلیدهای دسترسی^۲ به یک منو می توانید قبل از حرف مورد نظر در عنوان منو، یک علامت & قرار دهید. در مثال بالا حرف C کلید دسترسی به منو خواهد بود. کلیدهای میان بر^۳ فقط برای زیرمنوها قابل استفاده هستند. برای اینکه یک کلید میان بر به یک زیر منو نسبت دهید، باید خصوصیت Shortcut را مانند خصوصیت Caption دستکاری کنید. به خصوصیت Shortcut فقط می توان مقادیر مشخصی را نسبت داد. در ویژوال بیسیک

^۱ Collection

^۲ Access Key

^۳ Short-Cut Key

ثابت های خاصی مانند vbMenuAccelCtrlIO وجود دارند که به منظور استفاده در خصوصیت Shortcut بکار می روند.

برای ایجاد یک جداکننده در منو، ابتدا یک زیرمنو ایجاد کنید. بعد خصوصیت Caption را معادل خط فاصله (-) قرار دهید. این کار به ویژوال بیسیک اعلام می کند که شما در این مکان یک جداکننده منو نیاز دارید.

برای مخفی کردن یک آیتم از منو، خصوصیت Visible را مساوی با مقدار False قرار دهید. در مورد زیرمنوها، اگر تنها یک زیرمنوی قابل نمایش وجود داشته باشد نمی توان آنرا مخفی کرد.

۱- یک پروژه جدید در ویژوال بیسیک ایجاد کنید.

۲- فرم پیش فرض (Form1) را از پروژه حذف کنید.

۳- یک ماژول جدید به پروژه اضافه کنید (Module1.bas).

۴- کد زیر را در ماژول جدید وارد کنید:

```
''' MODULE1.BAS
Option Explicit

#If Win16 Then
    Declare Function WritePrivateProfileString Lib "KERNEL" ( _
        ByVal AppName As String, ByVal KeyName As String, _
        ByVal keydefault As String, ByVal FileName As String) As Integer
    Declare Function GetPrivateProfileString Lib "KERNEL" ( _
        ByVal AppName As String, ByVal KeyName As String, _
        ByVal keydefault As String, ByVal ReturnString As String, _
        ByVal NumBytes As Integer, ByVal FileName As String) As Integer
#ElseIf Win32 Then
    Declare Function WritePrivateProfileString Lib "Kernel32" _
        Alias "WritePrivateProfileStringA" (ByVal AppName As String, _
        ByVal KeyName As String, ByVal keydefault As String, _
        ByVal FileName As String) As Long
    Declare Function GetPrivateProfileString Lib "Kernel32" _
        Alias "GetPrivateProfileStringA" (ByVal AppName As String, _
        ByVal KeyName As String, ByVal keydefault As String, _
        ByVal ReturnString As String, ByVal NumBytes As Long, _
        ByVal FileName As String) As Long
#End If

Sub Main()
    #If Win16 Then
        Const Section = "Add-Ins16"
    #ElseIf Win32 Then
        Const Section = "Add-Ins32"
```

```

#End If

Const BufSize = 255
Dim Ret As Variant
Dim RetStr As String

' Hide the Addin from the Task Manager
App.TaskVisible = False

' Check to see if the entry is already in the VB.ini file.
' Add if not.
RetStr = Space(BufSize)
Ret = GetPrivateProfileString(Section, "AddinExample.MenuControl", _
    "NotFound", RetStr, BufSize, "VB.INI")
RetStr = Left(RetStr, Ret)
If RetStr = "NotFound" Then
    WritePrivateProfileString Section, "AddinExample.MenuControl", _
        "0", "VB.INI"
End If
End Sub

```

۵- یک ماژول از نوع کلاس به پروژه اضافه کنید (Class1.cls).

۶- خصوصیت های زیر را برای کلاس فوق تنظیم کنید:

مقدار	خصوصیت
1 - Creatable SingleUse	Instancing
MenuControl	Name
True	Public

۷- کد زیر را در کلاس فوق وارد کنید:

```

''' CLASS1.CLS
Option Explicit

Private ThisInstance As Object 'VBIDE.Application
Private AddInMenuLine As Object 'VBIDE.SubMenu
Private AddInID As Long

Public Sub ConnectAddIn(VBInstance As Object)
    Set ThisInstance = VBInstance
    Set AddInMenuLine = ThisInstance.AddInMenu.MenuItems.Add( _
        "Menu Control Example")
    AddInID = AddInMenuLine.ConnectEvents(Me)
End Sub

Public Sub DisconnectAddIn(Mode As Integer)
    AddInMenuLine.DisconnectEvents AddInID
    ThisInstance.AddInMenu.MenuItems.Remove AddInMenuLine
End Sub

Public Sub AfterClick()
    Dim CurrentForm As Object 'VBIDE.FormTemplate
    Dim mnuFile As Object 'VBIDE.ControlTemplate
    Dim mnuFileOpen As Object 'VBIDE.ControlTemplate

```

```

Dim mnuFileSeparator As Object 'VBIDE.ControlTemplate
Dim mnuFileMRU As Object 'VBIDE.ControlTemplate

' Set a reference to the Active Form
Set CurrentForm = ThisInstance.ActiveProject.ActiveForm

' Create the top-level File menu
Set mnuFile = CurrentForm.AddMenuTemplate("mnuFile", Nothing)
mnuFile.Properties("Caption").Value = "&File"

' Add the Open menu under the File menu
Set mnuFileOpen=CurrentForm.AddMenuTemplate("mnuFileOpen", mnuFile)
mnuFileOpen.Properties("Caption").Value = "&Open"
mnuFileOpen.Properties("Shortcut").Value = vbMenuAccelCtrlO

' Add a separator
Set mnuFileSeparator = CurrentForm.AddMenuTemplate _
    ("mnuFileSeparator", mnuFile)
mnuFileSeparator.Properties("Caption").Value = "-"

' Add MRU under the File menu
Set mnuFileMRU = CurrentForm.AddMenuTemplate("mnuFileMRU", mnuFile)
mnuFileMRU.Properties("Caption").Value = "MRU"
mnuFileMRU.Properties("Visible").Value = False ' Make Invisible
mnuFileMRU.Properties("Index").Value = 0 ' Make into control array
End Sub

```

۸- خصوصیت های زیر را برای پروژه تنظیم کنید:

مقدار	خصوصیت
Sub Main	Startup Form
AddinExample	Project Name
OLE Server	StartMode
Break in Class Module	Error Trapping

۹- پروژه را ذخیره کنید.

۱۰- برنامه را اجرا نمایید.

۱۱- یک بار دیگر ویژوال بیسیک را در پنجره دیگری اجرا کنید. Form1 بطور پیش فرض ایجاد می شود.

۱۲- به وسیله Add-in Manager ابزار AddinExample.MenuControl را به محیط اضافه کنید.

۱۳- تمرکز را روی Form1 قرار دهید (روی آن یکبار کلیک کنید).

۱۴- از منوی Add-in گزینه Menu Control Example را انتخاب کنید.

پس از این کار باید یک منوی File به Form1 اضافه شده باشد. در داخل این منو باید یک زیرمنوی Open وجود داشته باشد. این زیر منو باید کلید میان بر CTRL+O داشته باشد. زیر این گزینه Open یک جداکننده وجود دارد. زیر جداکننده یک آرایه از منوها که قابل دیدن نیست وجود خواهد داشت.

نمایش توضیحات در مورد آیتم های منو			
نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
185451.TXT	۱۵ مارچ ۱۹۹۹	5 , 6	95 / 98 / NT

بعضی از نرم افزارها مانند Explorer ویندوز این قابلیت را دارند که زمانی که آیتمی در منوهای آنها پررنگ می شود، توضیحاتی در مورد عملکرد آن آیتم در نوار وضعیت نمایش می دهند. این توانایی باعث می شود کاربر برنامه بتواند پیش از انتخاب گزینه های منو اطلاعات بیشتری در مورد نحوه عمل آن آیتم از منو بدست آورد. در مثال زیر نشان داده می شود که چگونه توضیحاتی برای آیتم های منو نمایش دهید. برای انجام این کار باید از مفهومی به نام کنترل پیام استفاده کنید. این عمل به شما امکان می دهد تشخیص دهید چه موقع پیام WM_MENUSELECT ارسال می شود. این پیام هر بار که آیتمی از منو پررنگ یا معمولی می شود ارسال می گردد. برای کسب اطلاعات بیشتر در مورد کنترل پیام می توانید به بخش شماره 168795 از بانک اطلاعاتی مایکروسافت رجوع کنید.

⚠️ **اخطار:** در صورتیکه پیش از پاک شدن پنجره از حافظه یا از بین رفتن آن عملیات کنترل پیام ها را متوقف نکنید، خطاهایی در برنامه بوجود آمده و باعث بهم ریختن سیستم شده و در نهایت اطلاعات شما از دست خواهد رفت. دلیل این اتفاق اینست که تابع WindowProc که


قرار است مورد استفاده قرار گیرد، قبلاً بخشی از برنامه بوده ولی دیگر وجود ندارد. همیشه پیش از خروج از برنامه یا پاک کردن پنجره از حافظه، به عملیات کنترل پیام خاتمه دهید. زمانی که در محیط ویژوال بیسیک به رفع اشکال چنین برنامه ای مشغول هستید، این نکته اهمیت بیشتری پیدا می کند. اگر پیش از خاتمه عملیات کنترل پیام، دکمه خاتمه برنامه را در محیط ویژوال بیسیک فشار دهید یا از منوی Run گزینه End را انتخاب کنید در مدیریت حافظه ویژوال بیسیک اشکالی بروز کرده و باعث بسته شدن ویژوال بیسیک می شود.

مراحل ایجاد مثال :

۱- یک پروژه معمولی ایجاد کنید.

۲- منوی زیر را توسط Menu Editor در Form1 ایجاد نمایید:

عنوان منو	نام منو
File	mnuFile
New	mnuNew
Open	mnuOpen
Close	mnuClose
Edit	mnuEdit
Cut	mnuCut
Copy	mnuCopy
Paste	mnuPaste
Normal	mnuPasteNormal
Special	mnuPasteSpecial

توجه:  لازمست عنوان منوها را دقیقاً مانند آنچه نشان داده شده است ایجاد کنید. اگر کلیدهای میان بر برای منو تعریف می کنید، لازمست بخشهای زیر از این مثال را طوری تغییر دهید که با عنوان منوهای شما سازگار باشند.

۳- کد زیر را در Form1 وارد کنید:

```
Option Explicit
```

```
Private Sub Form_Load()  
    'Store a handle to this form.  
    gHW = Me.hwnd
```

```
    'Call this Sub procedure to begin hooking into messages.
```

```
Hook
```

```

End Sub

Private Sub Form_Unload(Cancel As Integer)
    'Call this Sub procedure to cease hooking into messages.
    Unhook
End Sub

```

۴- یک ماژول معمولی به پروژه اضافه کنید.

۵- کد زیر را در ماژول وارد کنید:

```

Option Explicit

Private Declare Function CallWindowProc Lib "user32" _
    Alias "CallWindowProcA" (ByVal lpPrevWndFunc As Long, _
    ByVal hwnd As Long, ByVal Msg As Long, _
    ByVal wParam As Long, ByVal lParam As Long) As Long

Private Declare Function SetWindowLong Lib "user32" _
    Alias "SetWindowLongA" (ByVal hwnd As Long, _
    ByVal nIndex As Long, ByVal dwNewLong As Long) As Long

Private Declare Function GetMenu Lib "user32" _
    (ByVal hwnd As Long) As Long

Private Declare Function GetSubMenu Lib "user32" _
    (ByVal hMenu As Long, ByVal nPos As Long) As Long

Private Declare Function GetMenuItemCount Lib "user32" _
    (ByVal hMenu As Long) As Long

Private Declare Function GetMenuItemState Lib "user32" _
    (ByVal hMenu As Long, ByVal wID As Long, _
    ByVal wFlags As Long) As Long

Private Declare Function GetMenuString Lib "user32" _
    Alias "GetMenuStringA" (ByVal hMenu As Long, _
    ByVal wIDItem As Long, ByVal lpString As String, _
    ByVal nMaxCount As Long, ByVal wFlag As Long) As Long

Private Const MF_BYPOSITION = &H400&
Private Const MF_HILITE = &H80&
Private Const WM_MENUSELECT = &H11F
Private Const GWL_WNDPROC = -4

Public lpPrevWndProc As Long
Public gHW As Long

Public Sub Hook()
    'Begin hooking into messages.
    lpPrevWndProc = SetWindowLong(gHW, GWL_WNDPROC, _
    AddressOf WindowProc)
End Sub

Public Sub Unhook()
    'Cease hooking into messages.
    SetWindowLong gHW, GWL_WNDPROC, lpPrevWndProc
End Sub

```

```

Function AnyLit(hSubSubMenu As Long) As Long
    Dim i As Long
    Dim MenuCount As Long

    'Get the number of items in the menu.
    MenuCount = GetMenuItemCount(hSubSubMenu)

    'Loop through the menu items.
    For i = 0 To MenuCount - 1
        'Check whether this item is highlighted.
        If GetMenuState(hSubSubMenu, i, MF_BYPOSITION) And _
            MF_HILITE Then
            AnyLit = True
            Exit Function
        End If
    Next i

    'Return FALSE, no items highlighted.
    AnyLit = False
End Function

Private Sub WalkSubMenu(hSubMenu As Long)
    Dim i As Long
    Dim MenuItems As Long
    Dim hSubSubMenu As Long
    Dim buffer As String
    Dim result As Long

    'Get the count of menu items in this menu.
    MenuItems = GetMenuItemCount(hSubMenu)

    'Loop through all the items on the menu.
    For i = 0 To MenuItems - 1
        'Determine whether this item is highlighted.
        If GetMenuState(hSubMenu, i, MF_BYPOSITION) And _
            MF_HILITE Then
            'Attempt to get a submenu for each menu item.
            hSubSubMenu = GetSubMenu(hSubMenu, i)

            'Check for a submenu with something selected on it.
            If hSubSubMenu And AnyLit(hSubSubMenu) Then
                'There is a submenu with a selection so walk it.
                WalkSubMenu hSubSubMenu
            Else
                'This is it.
                'Set buffer size.
                buffer = Space(255)

                'Call the API to get the caption for the menu item.
                result = GetMenuString(hSubMenu, i, buffer, _
                    Len(buffer), MF_BYPOSITION)

                'Trim the buffer of extra characters.
                buffer = Left$(buffer, result)

                'Set the caption of the form to a description of the
                'menu item.
                Form1.Caption = GetDescription(buffer)

                'Exit this Sub procedure.
            End If
        End If
    Next i
End Sub

```

```
        Exit Sub
    End If
End If
Next i
End Sub

Public Sub FindHilite(TheForm As Form)
    Dim hMenu As Long
    Dim hSubMenu As Long
    Dim i As Long
    Dim MenuCount As Long

    'Clear any previous description.
    Form1.Caption = ""

    'Get the menu handle.
    hMenu = GetMenu(TheForm.hwnd)

    'Check to see if there is no menu.
    If hMenu <> 0 Then
        'Get the number of top-level menus.
        MenuCount = GetMenuItemCount(hMenu)

        'Enumerate through all top-level menus.
        For i = 0 To MenuCount - 1
            'Ignore top-level menus not currently selected.
            If GetMenuState(hMenu, i, MF_BYPOSITION) And _
                MF_HILITE Then
                'Get a handle to the submenu.
                hSubMenu = GetSubMenu(hMenu, i)

                'Walk the submenu.
                WalkSubMenu hSubMenu
            End If
        Next i
    End If
End Sub

Private Function GetDescription(MenuCaption As String) As String
    'Determine the description of the menu item.
    Select Case MenuCaption
        Case "New"
            GetDescription = "Creates a new document"
        Case "Open"
            GetDescription = "Opens a Document"
        Case "Close"
            GetDescription = "Closes Document"
        Case "Cut"
            GetDescription = "Cuts Selection to Clipboard"
        Case "Copy"
            GetDescription = "Copies Selection to Clipboard"
        Case "Paste"
            GetDescription = "Pastes Contents of Clipboard"
        Case "Normal"
            GetDescription = "Regular Paste"
        Case "Special"
            GetDescription = "Special Paste"
        Case Else
            GetDescription = ""
    End Select
End Function
```



```

End Select
End Function

Function WindowProc(ByVal hw As Long, ByVal uMsg As Long, _
    ByVal wParam As Long, ByVal lParam As Long) As Long

    'Check for a menu selection message.
    If uMsg = WM_MENUSELECT Then
        FindHilite Form1
    End If

    'Pass the message to Windows for processing.
    WindowProc = CallWindowProc(lpPrevWndProc, hw, uMsg, _
        wParam, lParam)
End Function

```

۶- برای اجرای برنامه از کلید F5 یا دکمه Start استفاده کنید. منوهای مختلف برنامه را باز کرده و تغییراتی را که در عنوان پنجره بوجود می آید زیر نظر داشته باشید.

جلوگیری از نمایش منوی pop-up پیش فرض

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
95 / 98 / NT	5 , 6	۲۴ آگوست ۱۹۹۸	191670.TXT

بعضی از ابزارهای ویژوال بیسیک مانند TextBox دارای منوی Pop-up پیش فرضی هستند که زمانی که کاربر روی آنها با دکمه راست ماوس کلیک می کند بطور خودکار ظاهر می شوند. در این بخش روشی را خواهیم دید که بوسیله آن از ظاهر شدن این منو جلوگیری می شود تا بتوان منوی دلخواه دیگری را به جای آن نمایش داد.

هنگامی که روی یک TextBox با دکمه سمت راست ماوس کلیک می کنید، منوی Pop-up پیش فرض آن نمایش داده می شود. ویژوال بیسیک هیچ خصوصیت یا مکانیزم داخلی دیگری ندارد تا این خاصیت را از کار بیاندازد. با این حال، تغییر خصوصیت Enabled این ابزار به حالت False از نمایش داده شدن این منو جلوگیری می کند. البته، کاربر برنامه نیز متوجه خواهد شد که ابزار مورد بحث غیر فعال شده است.

یک روش برای مقابله با این مشکل اینست که از تابع LockWindowUpdate به همراه خصوصیت Enabled استفاده شود. این تابع ترسیمات روی یک پنجره مشخص را از کار انداخته و یا دوباره فعال می کند. بعد از اینکه عملیات کامل شد، ابزار فعال شده و تابع LockWindowUpdate فراخوانده می شود تا عملیات ترسیم شدن ابزار را دوباره بکار اندازد.

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

۲- یک TextBox به Form1 اضافه کنید.

۳- با استفاده از ویرایشگر منو در پنجره Form1 یک منو به نام mnuPopUp ایجاد کنید. خصوصیت Visible آنرا غیر فعال کنید و آیتم هایی مانند اینها به منو اضافه کنید:

```
File
  New
  Open
```

۴- کد زیر را به Form1 اضافه کنید:

```
Private Declare Function LockWindowUpdate Lib "user32" _
    (ByVal hwndLock As Long) As Long

Private Sub mnuOne_Click()
    Text1.Text = "Menu One was clicked"
End Sub

Private Sub mnuTwo_Click()
    Text1.Text = "Menu two was clicked"
End Sub

Private Sub Text1_MouseDown(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)

    If Button = vbRightButton Then
        ' Avoid the 'disabled' gray text by locking updates
        LockWindowUpdate Text1.hWnd

        ' A disabled TextBox will not display a context menu
        Text1.Enabled = False

        ' Give the previous line time to complete
        DoEvents

        ' Display our own context menu
        PopupMenu mnuPopup

        ' Enable the control again
```

```

Text1.Enabled = True
' Unlock updates
LockWindowUpdate 0&
End If
End Sub

```

۵- برنامه را ذخیره کرده و اجرا نمایید.

۶- روی Text1 با دکمه سمت راست کلیک کنید. فقط منوی جدید نمایش داده می شود.

منوی ویرایش استاندارد نمایش داده نخواهد شد.

روش دیگری که برای جلوگیری از نمایش منوی Pop-up پیش فرض وجود دارد، کنترل پیام های ارسالی از ویندوز است. از طریق کنترل پیام، می توانید پیام های خاصی از ماوس را کنترل کرده و خودتان آنها را پردازش کنید. برای اطلاعات بیشتر می توانید به مراجع زیر مراجعه کنید:

ARTICLE-ID: Q170570
TITLE : HOWTO: Build a Windows Message Handler with AddressOf in VB

ARTICLE-ID: Q155969
TITLE : HOWTO: Distribute the WebBrowser Control

ایجاد منوهای چند ستونی در ویژوال بیسیک توسط توابع WIN32

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
216185.TXT	۴ فوریه ۱۹۹۹	5 , 6	95 / 98 / NT

زمانی که نمی خواهید بخشهایی از پنجره با منوها پوشیده شود یا زمانی که منوهای بسیار بزرگی دارید که ممکن است از صفحه بیرون بزند، استفاده از منوهای چند ستونی می تواند مفید باشد. ویژوال بیسیک خودش به تنهایی روشی برای ایجاد منوهای چند ستونی در اختیار برنامه نویسان نمی گذارد، اما این امکان وجود دارد که یک منو را از طریق فراخوانی توابع

API بصورت چند ستونی نمایش دهید. مثال زیر نشان می دهد که این کار چگونه قابل انجام است.

توجه: این تکنیک فقط برای منوهای که قابل نمایش روی نوار منو هستند کاربرد دارد و

برای منوهای غیر قابل دیده شدن مانند منوهای Pop-up عملی نیست.

۱- یک پروژه جدید ایجاد کنید. Form1 بطور پیش فرض ایجاد می شود.

۲- یک CommandButton به Form1 اضافه کنید.

۳- از منوی Tools گزینه Menu Editor را انتخاب کنید. منویی با حداقل دو عنوان

اصلی و حداقل چهار آیتم برای هر یک از دو عنوان اصلی ایجاد کنید.

۴- کد زیر را به بخش Declarations از Form1 اضافه نمایید.

Option Explicit

Private Type MENUITEMINFO

cbSize As Long

fMask As Long

fType As Long

fState As Long

wID As Long

hSubMenu As Long

hbmChecked As Long

hbmUnchecked As Long

dwItemData As Long

dwTypeData As String

cch As Long

End Type

Private Const MF_MENUBARBREAK = &H20& ' columns with a separator line

Private Const MF_MENUBREAK = &H40& ' columns w/o a separator line

Private Const MF_STRING = &H0&

Private Const MF_HELP = &H4000&

Private Const MFS_DEFAULT = &H1000&

Private Const MIIM_ID = &H2

Private Const MIIM_SUBMENU = &H4

Private Const MIIM_TYPE = &H10

Private Const MIIM_DATA = &H20

Private Declare Function GetMenu Lib "user32" (ByVal hwnd As Long) _
As Long

Private Declare Function GetMenuItemInfo Lib "user32" _
Alias "GetMenuItemInfoA" _
(ByVal hMenu As Long, ByVal un As Long, ByVal B As Boolean, _
lpMenuItemInfo As MENUITEMINFO) As Long

Private Declare Function SetMenuItemInfo Lib "user32" _
Alias "SetMenuItemInfoA" _

```

        (ByVal hMenu As Long, ByVal un As Long, ByVal bool As Boolean, _
        lpMenuItemInfo As MENUITEMINFO) As Long

Private Declare Function DrawMenuBar Lib "user32" _
    (ByVal hwnd As Long) As Long

Private Declare Function GetSubMenu Lib "user32" _
    (ByVal hMenu As Long, ByVal nPos As Long) As Long

Private Sub Command1_Click()
    ' Splitting a menu here demonstrates that this can be done dynamically.
    Dim mnuItemInfo As MENUITEMINFO, hMenu As Long, hSubMenu As Long
    Dim BuffStr As String * 80 ' Define as largest possible menu text.

    hMenu = GetMenu(Me.hwnd) ' retrieve menu handle.
    BuffStr = Space(80)
    With mnuItemInfo ' Initialize the UDT.
        .cbSize = Len(mnuItemInfo) ' 44
        .dwTypeData = BuffStr & Chr(0)
        .fType = MF_STRING
        .cch = Len(mnuItemInfo.dwTypeData) ' 80
        .fState = MFS_DEFAULT
        .fMask = MIIM_ID Or MIIM_DATA Or MIIM_TYPE Or MIIM_SUBMENU
    End With
    ' Use item break point position for the '3' below (zero-based list).
    hSubMenu = GetSubMenu(hMenu, 0)
    If GetMenuItemInfo(hSubMenu, 2, True, mnuItemInfo) = 0 Then
        MsgBox "GetMenuItemInfo failed. Error: " & Err.LastDllError, , _
            "Error"
    Else
        mnuItemInfo.fType = mnuItemInfo.fType Or MF_MENUBARBREAK
        If SetMenuItemInfo(hSubMenu, 2, True, mnuItemInfo) = 0 Then
            MsgBox "SetMenuItemInfo failed. Error: " & Err.LastDllError, , _
                "Error"
        End If
    End If
    DrawMenuBar (Me.hwnd) ' Repaint top level Menu.
End Sub

Private Sub Form_Load()
    ' This works for either an API-created menu or a native VB Menu.
    Dim mnuItemInfo As MENUITEMINFO, hMenu As Long, hSubMenu As Long
    Dim BuffStr As String * 80 ' Define as largest possible menu text.

    hMenu = GetMenu(Me.hwnd) ' Retrieve menu handle.
    BuffStr = Space(80)
    With mnuItemInfo ' Initialize the UDT
        .cbSize = Len(mnuItemInfo) ' 44
        .dwTypeData = BuffStr & Chr(0)
        .fType = MF_STRING
        .cch = Len(mnuItemInfo.dwTypeData) ' 80
        .fState = MFS_DEFAULT
        .fMask = MIIM_ID Or MIIM_DATA Or MIIM_TYPE Or MIIM_SUBMENU
    End With
    ' Use item break point position for the '3' below (zero-based list).
    hSubMenu = GetSubMenu(hMenu, 1)
    If GetMenuItemInfo(hSubMenu, 2, True, mnuItemInfo) = 0 Then
        MsgBox "GetMenuItemInfo failed. Error: " & Err.LastDllError, , _
            "Error"
    End If
End Sub

```

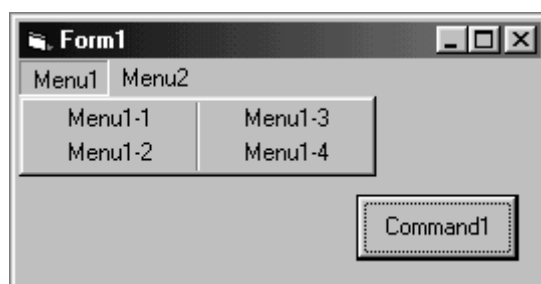
```

Else
    mnuItemInfo.fType = mnuItemInfo.fType Or MF_MENUBREAK
    If SetMenuItemInfo(hSubMenu, 2, True, mnuItemInfo) = 0 Then
        MsgBox "SetMenuItemInfo failed. Error: " & Err.LastDllError, , _
            "Error"
    End If
End If
End If
DrawMenuBar (Me.hwnd) ' Repaint top level Menu.
End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
    If Button = vbRightButton Then
        PopupMenu Aaa ' Use the name of one of your menus.
    End If
End Sub
End Sub

```


۵- عبارت Aaa در آخرین روتین بالا را به نام یکی از منوهای اصلی خود تغییر دهید. نتیجه این خواهد بود که اولین منو مطابق معمول باز می شود، اما دومین منو دو ستون دارد. به علاوه، کلیک کردن با دکمه سمت راست ماوس باعث می شود که منوی اول به صورت یک منوی Pop-up ظاهر شود. اکنون روی CommandButton کلیک کنید و دوباره آزمایش های بالا را انجام دهید. این بار منوی اول و منوی pop-up دو ستون دارند که بایک خط جدا شده اند. اگر بخواهید بطور دینامیکی ستونها را تقسیم کنید باید از تابع GetMenuItemCount برای تعیین تعداد آیتمهای منو استفاده کرده و سپس توسط یک حلقه آنها را پردازش کنید، به این ترتیب که برای تعیین محل تقسیم منو به دنبال یک رشته خاص در عنوان آنها بگردید یا فقط به ازای هر X آیتم یک ستون جدید ایجاد کنید.



شکل 1 - منوهای چند ستونی با خط عمودی جدا کننده

قراردادن منوی Help در سمت راست نوار منو

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
216189.TXT	۵ فوریه ۱۹۹۹	5 , 6	95 / 98 / NT

در بعضی از برنامه ها (خصوصا برنامه های قدیمی تر ۱۶ بیتی) منوی Help در سمت راست نوار منو قرار دارد، یعنی به جای اینکه در کنار دیگر منوها در سمت چپ باشد، به تنهایی در سمت راست پنجره قرار می گیرد. ویرایشگر منوی ویژوال بیسیک این امکان را در اختیار قرار نمی دهد، اما به کمک چند تابع API می توان این کار را به راحتی انجام داد.  توجه: قرار دادن منوی راهنما در سمت راست یک برنامه ویندوز توصیه نمی شود.

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

۲- در منوی Tools گزینه Menu Editor را انتخاب کنید. حداقل چهار منوی اصلی ایجاد کنید که آخرین آنها عنوان Help داشته باشد.

۳- کد زیر را در بخش Declarations از Form1 وارد کنید:

```
Option Explicit

Private Type MENUITEMINFO
    cbSize As Long
    fMask As Long
    fType As Long
    fState As Long
    wID As Long
    hSubMenu As Long
    hbmpChecked As Long
    hbmpUnchecked As Long
    dwItemData As Long
    dwTypeData As String
    cch As Long
End Type

Private Const MF_STRING = &H0&
Private Const MF_HELP = &H4000&
Private Const MFS_DEFAULT = &H1000&

Private Const MIIM_ID = &H2
Private Const MIIM_SUBMENU = &H4
Private Const MIIM_TYPE = &H10
Private Const MIIM_DATA = &H20
```

```

Private Declare Function GetMenu Lib "user32" _
    (ByVal hwnd As Long) As Long

Private Declare Function GetMenuItemInfo Lib "user32" _
    Alias "GetMenuItemInfoA" _
    (ByVal hMenu As Long, ByVal un As Long, ByVal B As Boolean, _
    lpMenuItemInfo As MENUITEMINFO) As Long

Private Declare Function SetMenuItemInfo Lib "user32" _
    Alias "SetMenuItemInfoA" _
    (ByVal hMenu As Long, ByVal un As Long, ByVal bool As Boolean, _
    lpMenuItemInfo As MENUITEMINFO) As Long

Private Declare Function DrawMenuBar Lib "user32" _
    (ByVal hwnd As Long) As Long

Private Sub Form_Load()
    Dim mnuItemInfo As MENUITEMINFO, hMenu As Long
    Dim BuffStr As String * 80 ' Define as largest possible menu text.

    hMenu = GetMenu(Me.hwnd) ' Retrieve the menu handle.
    BuffStr = Space(80)
    With mnuItemInfo ' Initialize UDT
        .cbSize = Len(mnuItemInfo) ' 44
        .dwTypeData = BuffStr & Chr(0)
        .fType = MF_STRING
        .cch = Len(mnuItemInfo.dwTypeData) ' 80
        .fState = MFS_DEFAULT
        .fMask = MIIM_ID Or MIIM_DATA Or MIIM_TYPE Or MIIM_SUBMENU
    End With
    'Use the desired item's position for the '3' below (zero-based list).
    If GetMenuItemInfo(hMenu, 3, True, mnuItemInfo) = 0 Then
        MsgBox "GetMenuItemInfo failed. Error: " & Err.LastDllError, , _
            "Error"
    Else
        mnuItemInfo.fType = mnuItemInfo.fType Or MF_HELP
        If SetMenuItemInfo(hMenu, 3, True, mnuItemInfo) = 0 Then
            MsgBox "SetMenuItemInfo failed. Error: " & Err.LastDllError, , _
                "Error"
        End If
    End If
    DrawMenuBar (Me.hwnd) ' Repaint top level Menu
End Sub

```

۴- برنامه را اجرا کنید. توجه کنید که منوی Help در سمت راست نوار منو قرار گرفته است و کاملاً طبیعی عمل می کند.



شکل 2 - قرار دادن منو در سمت راست

قراردادن تصاویر گرافیکی در منوها			
---	--	--	--

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
3.1	1 , 2 , 3	۲۱ ژانویه ۱۹۹۵	71281.TXT

هیچ فرمانی در ویژوال بیسیک وجود ندارد که بتوان به کمک آن Bitmap هایی را به منوهای برنامه اضافه نمود، اما می توانید با فراخوانی چند تابع API این کار را انجام دهید. حتی می توانید علامت چک مارک کنار منو را نیز تغییر دهید.

توابعی که در زیر معرفی می شوند برای نمایش Bitmap روی منوها بکار می روند:

– `GetMenu% (hwnd%)`

`hwnd%` : شناسه پنجره ای که منوهای روی آن مورد بررسی قرار می گیرد

مقدار بازگردانده شده : شناسه منوی مورد نظر

– `GetSubMenu% (hMenu%, nPos%)`

`hMenu%` : شناسه منو

`nPos%` : مشخص کننده مکان زیرمنوی مورد نظر در منوی فوق (از صفر شروع

می شود)

مقدار بازگردانده شده : شناسه زیرمنوی مورد نظر

– `GetMenuItemID% (hMenu%, nPos%)`

`hMenu%` : شناسه منویی که حاوی آیتم مورد نظر است

`nPos%` : مشخص کننده مکان آیتمی که شناسه اش مورد نظر است (از صفر شروع

می شود)

مقدار بازگردانده شده : شناسه آیتم مورد نظر

– `ModifyMenu% (hMenu%, nPos%, wFlags%, wIDNewItem%, lpNewItem&)`

`hMenu%` : شناسه منویی که آیتم مورد نظر در آن قرار دارد

nPos% : مشخص کننده آیتمی که قرار است تغییر کند. تفسیر مقدار nPos به پارامتر wFlags بستگی دارد

WFlags% : برای این مثال باید معادل BF_BITMAP (که مقدار &H4 دارد) باشد

wIDNewItem% : مشخص کننده شناسه آیتمی است که قرار است دستکاری شود

lpNewItem& : شناسه ۳۲ بیتی Bitmap

مقدار بازگردانده شده : True در صورت موفقیت تابع و False در صورت عدم موفقیت

- SetMenuItemBitmaps% (hMenu%, nPos%, Flags%, hBitmapUnchecked%, hBitmapChecked%)

hMenu% : شناسه منویی که آیتم مورد نظر در آن قرار دارد

nPos% : مشخص کننده شناسه آیتمی است که قرار است دستکاری شود

&H0 : WFlags%

hBitmapUnchecked% : شناسه بیت مپ حالت بدون چک مارک

hBitmapChecked% : شناسه بیت مپ چک مارک

مقدار بازگردانده شده : True در صورت موفقیت تابع و False در صورت عدم موفقیت

دو راه متفاوت برای استفاده از Bitmap ها در ویژوال بیسیک وجود دارد: روش اول استفاده

از Bitmap های ایستا یا استاتیک و روش دیگر استفاده از Bitmap پویا یا دینامیک است.

Bitmap ایستا ثابت است و در طول اجرای برنامه تغییر نمی کند (مانند آنهایی که از

فایلهای BMP گرفته می شوند). Bitmap های پویا در طول اجرای برنامه تغییر می کنند. در

Bitmap پویا ممکنست خصوصیتی مانند رنگ، اندازه و متن نوشته شده روی آنها تغییر کند.

مثال زیر روش ایجاد هر دو نوع منو را نشان می دهد:

در برنامه مورد نظر خود یک سیستم منو توسط پنجره طراحی منو ایجاد کنید و ساختار

زیر را در آن بوجود آورید:

عنوان	نام منو	زیر منو است	Index
BitMenu	TopMenu	خیر	
Sub Menu0	SubMenu	بله (یک مرحله)	0
Sub Menu1	SubMenu	بله (یک مرحله)	1
Sub Menu2	SubMenu	بله (یک مرحله)	2

یک آرایه از PictureBox برای ۳ Bitmap ایجاد کنید (نام همگی را Picture1 اختیار کنید). Picture1 برای ذخیره Bitmap های منوها بکار می رود. یک PictureBox دیگر بنام Picture2 برای ذخیره Bitmap چک مارک ایجاد کنید. یک آرایه ۳ عضوی از نوع PictureBox نیز برای سایز فونتهای مختلف ایجاد کنید و خصوصیت سایز فونت هر یک از آنها را متفاوت با دیگری اختیار کنید. دو CommandButton نیز روی فرم ایجاد کنید. خصوصیت های این ابزارها باید بصورت جدول زیر تنظیم شوند:

نام	عنوان	Index	سایز فونت
Picture1		0	مهم نیست
Picture1		1	مهم نیست
Picture1		2	مهم نیست
Picture2		ندارد	مهم نیست
Picture3		0	8
Picture3		1	9.75
Picture3		2	18
Command1	Static		
Command2	Dynamic		

به هر یک از اعضای آرایه Picture1 یک Bitmap نسبت دهید. از آنجا که این Bitmap ها قرار است در منو نمایش داده شوند باید از تصاویر کوچک استفاده نمایید. یک Bitmap هم به Picture2 نسبت دهید تا از آن به عنوان علامت چک مارک استفاده شود.
در هر دو روش ایستا و پویا باید کد زیر را در برنامه خود داشته باشید:

```
' Enter each Declare statement on one, single line:
Declare Function GetMenu% Lib "user" (ByVal hwnd%)
Declare Function GetSubMenu% Lib "user" (ByVal hMenu%, ByVal nPos%)
Declare Function GetMenuItemID% Lib "user" (ByVal hMenu%, ByVal nPos%)
Declare Function ModifyMenu% Lib "user" (ByVal hMenu%, ByVal nPosition%,
    ByVal wFlags%, ByVal wIDNewItem%, ByVal lpNewItem%)
Declare Function SetMenuItemBitmaps% Lib "user" (ByVal hMenu%,
    ByVal nPosition%, ByVal wFlags%, ByVal hBitmapUnchecked%,
    ByVal hBitmapChecked%)
Const MF_BITMAP = &H4
Const CLR_MENUBAR = &H80000004 ' Defined for dynamic bitmaps only.
Const TRUE = -1, FALSE = 0
Const Number_of_Menu_Selections = 3
```

سابروتین زیر نیز باید برای کنترل وضعیت علامت چک مارک در برنامه بکار رود. کل کد

زیر را در برنامه وارد کنید:

```
Sub SubMenu_Click (Index As Integer)
' Uncheck presently checked item, check new item, store
' index
    Static LastSelection%
    SubMenu(LastSelection%).Checked = FALSE
    SubMenu(Index).Checked = TRUE
    LastSelection% = Index
End Sub

Sub Command1_Click ()
' * example to create a static bitmap menu
hMenu% = GetMenu(hwnd)
hSubMenu% = GetSubMenu(hMenu%, 0)
For i% = 0 To Number_of_Menu_Selections - 1
    menuItem% = GetMenuItemID(hSubMenu%, i%)
    x% = ModifyMenu(hMenu%, menuItem%, MF_BITMAP, menuItem%,
        CLng(Picture1(i%).Picture))
    x% = SetMenuItemBitmaps(hMenu%, menuItem%, 0, 0,
        CLng(Picture2.Picture))
Next i%
End Sub

' This code sample will change the actual menu bitmaps size,
' font size, color, and caption. Run the application and
' select the BitMenu and view the selections. Then click
' the form and revisit the BitMenu.
'-----
Sub Command2_Click ()
' * Example to create a dynamic menu system
hMenu% = GetMenu(hwnd)
hSubMenu% = GetSubMenu(hMenu%, 0)
For i% = 0 To Number_of_Menu_Selections - 1
' * Place some text into the menu.

    SubMenu(i%).Caption = Picture3(i%).FontName +
        Str$(Picture3(i%).FontSize) + " Pnt"

' * 1. Must be AutoRedraw for Image().
' * 2. Set Backcolor of Picture control to that of the
' * current system Menu Bar color, so Dynamic bitmaps
```

```

'* will appear as normal menu items when menu bar
'* color is changed via the control panel
'* 3. See the bitmaps on screen, this could all be done
'* at design time.

Picture3(i%).AutoRedraw = TRUE
Picture3(i%).BackColor = CLR_MENUBAR
'* You can uncomment this
'* Picture3(i%).Visible = FALSE

'* Set the width and height of the Picture controls
'* based on their corresponding Menu items caption,
'* and the Picture controls Font and FontSize.
'* DoEvents() is necessary to make new dimension
'* values to take affect prior to exiting this Sub.

Picture3(i%).Width = Picture3(i%).TextWidth(SubMenu(i%).Caption)
Picture3(i%).Height = Picture3(i%).TextHeight(SubMenu(i%).Caption)
Picture3(i%).Print SubMenu(i%).Caption

'* - Set picture controls backgroup picture (Bitmap) to
'* its Image.
Picture3(i%).Picture = Picture3(i%).Image
x% = DoEvents()
Next i%

'* Get handle to forms menu.
hMenu% = GetMenu(Form1.hWnd)

'* Get handle to the specific menu in top level menu.
hSubMenu% = GetSubMenu(hMenu%, 0)

For i% = 0 To Number_of_Menu_Selections - 1

'* Get ID of sub menu
menuId% = GetMenuItemID(hSubMenu%, i%)

'* Replace menu text w/bitmap from corresponding picture
'* control
x% = ModifyMenu(hMenu%, menuId%, MF_BITMAP, menuId%,
CLng(Picture3(i%).Picture)) 'append this to previous line

'* Replace bitmap for menu check mark with custom check
'* bitmap
x% = SetMenuItemBitmaps(hMenu%, menuId%, 0, 0, CLng(picture2.Picture))
Next i%
End Sub

```

برای اجرای مثال فوق در ویندوز ۹۵ به بالا و ویژوال بیسیک ۵ باید در بخش تعریف توابع

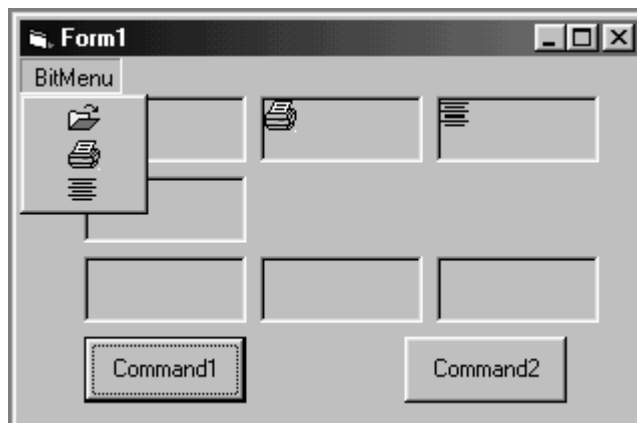
API عبارت "user" را به "user32" تغییر داده و نیز تابع ModifyMenu را بصورت زیر تعریف

نمایید:

```

Declare Function ModifyMenu Lib "user32" Alias "ModifyMenuA" _
(ByVal hMenu As Long, ByVal nPosition As Long, _
ByVal wFlags As Long, ByVal wIDNewItem As Long, _
ByVal lpString As Any) As Long

```



شکل 3 - تصاویر گرافیکی در منو



شکل 4 - استفاده از فونت‌های مختلف در منو