فصل دوم

چند رسانه ای

هدف‌های رفتاری: فراگیر پس از پایان این فصل، خواهد توانست :

۱ــ از کنترل‌ها و ابزارهای چند رسانه‌ای در تولید برنامه‌ها استفاده کند ؛

۲ــ فایل‌های صوتی و ویدیویی را در محیط یک برنامه پخش و نمایش دهد ؛

۳ــ پروژه‌های چند رسانه‌ای را طراحی و پیاده‌سازی کند.

۱ــ۲ــ کنترل چند رسانه‌ای

کنترل چند رسانه‌ای (multimedia) علیرغم قدرتش، بسیار ساده است. برای عملیاتی کردن این کنترل فقط مقدار کمی کدنویسی لازم است. با کنترل چند رسانه‌ای می‌توانید از ابزارهای چند رسانه‌ای زیر استفاده کنید.

- پخش CD صوتی (CD Audio)
- پخش نوار صوتی دیجیتالی (DAT)
- پخش فایل‌های ویدیویی دیجیتالی (Digital Video)
- (اسکنرScanner)
- ضبط و پخش نوار ویدیو (VCR)
- پخش دیسک ویدیویی (Video Disc)
- ابزارهای دیگر (Other)

(مقادیری که در پرانتز می‌بینید مربوط به مشخصه‌ی Device Name کنترل چند رسانه‌ای هستند.) علاوه بر ابزارهای فوق، کنترل چند رسانه‌ای می‌تواند با شیءهای زیر هم کار کند :

- فایل‌های صوتی (. WAV)

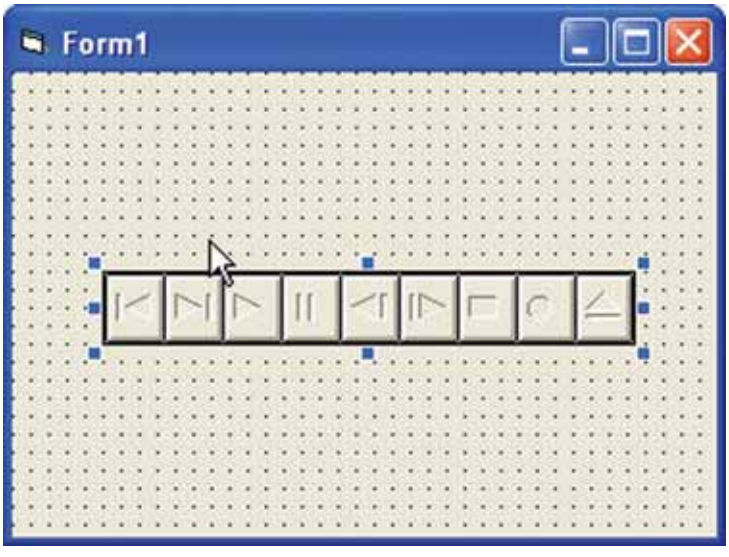
- فایل‌های صوتی ــ تصویری (.AVI)

به ابزارهایی که برای کار به فایل نیاز ندارند، ابزارهای ساده‌ی چند رسانه‌ای می‌گویند و ابزارهایی که برای کار خود به فایل نیاز دارند، ابزارهای مرکب چند رسانه‌ای نامیده می‌شوند. مثلاً CD صوتی یک ابزار ساده است چون برای پخش آن نیاز به هیچ فایلی ندارید (کافی است CD را درون درایو قرار داده و گوش کنید)، اما برای پخش فایل‌های صوتی به فایل‌های WAV. نیاز دارید و ابزار مرکب محسوب می‌شوند.

نیازی به توضیح نیست که کامپیوتری که می‌خواهد با ابزارهای چند رسانه‌ای کار کند باید ابزارهای لازمه (از قبیل درایو CD، کارت صوتی و کارت ویدیویی مناسب) را داشته باشد.

۱ــ۱ــ۲ــ کار با کنترل چند رسانه‌ای

برای استفاده از چند رسانه‌ای باید کنترل Microsoft Multimedia Control 6.0 را به جعبه ابزار خود اضافه کرده باشید. بعد از دابل کلیک روی ابزار چند رسانه‌ای، این کنترل، با دکمه‌های آشنای خود، روی فرم ظاهر خواهد شد (شکل ۱ــ۲).

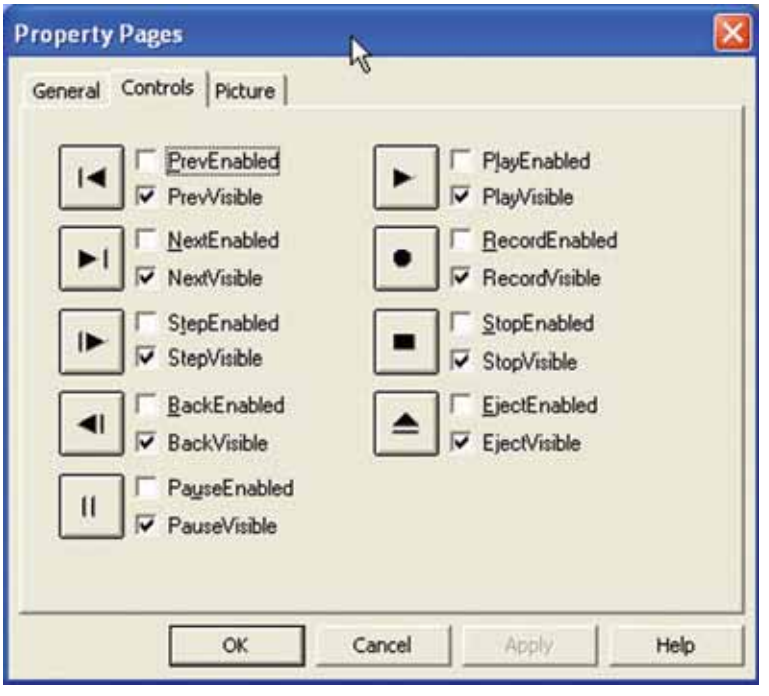


شکل ۱ــ۲ــ کنترل چند رسانه‌ای و دکمه‌های آن

کنترل چند رسانه‌ای می‌تواند قابلیت‌های ابزاری که به آن وصل شده است را تشخیص دهد. به عنوان مثال، اگر CD درون درایو قرار نداشته باشد دکمه‌ی پخش (►) فعال نخواهد شد. برای کار

با این کنترل هم باید از مشخصه‌های آن استفاده کنید. کنترل چند رسانه‌ای حتی اطلاعاتی را که از ابزار متصل به آن گرفته است، به برنامه انتقال می‌دهد. (به عنوان مثال، اگر در حال پخش CD صوتی باشد، می‌تواند شماره‌ی Track را به برنامه بدهد).

وقتی به مشخصه DeviceType کنترل چند رسانه‌ای مقداری می‌دهید، این کنترل متناسب با ابزار انتخاب شده، دکمه‌های لازم را فعال خواهد کرد ؛ بنابراین شما نباید نگران دکمه‌های این کنترل باشید. کنترل چند رسانه‌ای هم مانند سایر کنترل‌های پیچیده دارای مشخصه‌ای به نام Custom است که تنظیم مشخصه‌های آن را ساده‌تر خواهد کرد. شکل ۲ــ۲ زبانه‌ی Controls صفحات این کنترل را نشان می‌دهد. در این صفحه می‌توانید دکمه‌های مورد نیاز را فعال یا مرئی کنید.



شکل ۲ــ۲ــ صفحات مشخصه‌های کنترل چند رسانه‌ای

۲ــ۱ــ۲ــ پخش CD صوتی

برای ایجاد یک برنامه‌ی پخش CD صوتی، باید یک کنترل چند رسانه‌ای روی فرم برنامه قرار داده و مشخصه‌ی DeviceType آن را با CDAudio مقداردهی کنید. تا همین‌جا هم یک برنامه‌ی

عملیاتی دارید، اما اگر می‌خواهید شماره‌ی Track در حال پخش را هم مشاهده کنید، باید یک برچسب روی فرم قرار دهید و در روال رویداد StatusUpdate شماره‌ی Track را به هنگام کنید.

۳ــ۱ــ۲ــ زبان کنترل چند رسانه‌ای

کنترل چندرسانه‌ای یک زبان خاص خود را دارد و فرمان‌های خودر ا از طریق مشخصه Command می‌گیرد. در جدول ۱ــ۲ فرامین این کنترل را مشاهده می‌کنید.

جدول ۱ــ۲ــ فرامین کنترل چند رسانه‌ای

فرمان	مفهوم
Back	یک Track به عقب.
Close	ابزار را می‌بندد.
Eject	CD را از درایو بیرون می‌دهد.
Next	یک Track به جلو (اگر آخرین Track باشد، به ابتدای همان Track برمی‌گردد).
Open	ابزار را باز می‌کند.
Pause	توقف موقت ابزار (مکث).
Play	پخش ابزار.
Prev	به ابتدای Track جاری می‌رود. (اگر در ۳ ثانیه اول Track باشد، به ابتدای Track قبلی خواهد رفت.)
Record	شروع عملیات ضبط.
Save	فایل ابزار را ذخیره می‌کند.
Seek	یک Track به جلو یا عقب. (اغلب برنامه‌نویسان از Next یا Prev به جای این فرمان استفاده می‌کنند.)
Stop	توقف ابزار.
Step	حرکت در Track فعلی به جلو.

می‌توان کنترل چند رسانه‌ای را پنهان کرده و با استفاده از فرمان‌های فوق آن را کنترل کرد، چون این کنترل بلافاصله به فرمان‌ها جواب می‌دهد.

مثال ۱ــ۲ــ حال که با کنترل چند رسانه‌ای آشنا شدید، اجازه دهید یک برنامه‌ی پخش CD بنویسیم. ابتدا، با استفاده از جدول ۲ــ۲ فرم برنامه را بسازید.

جدول ۲ــ۲ــ مشخصه‌های کنترل برنامه‌ی پخش CD

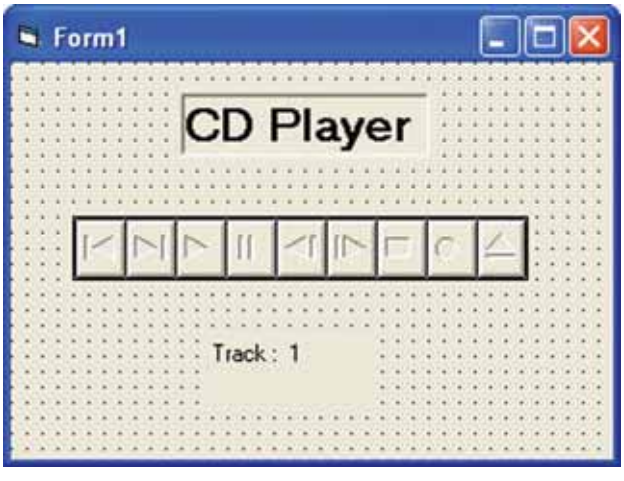
کنترل	مقدار
Form Name	FrmCD
Form Caption	Cd Player
Form Height	3600
Form Width	4800
Label#1 Name	lblCD
Label#1 Alignment	2-Center
Label# 1 BorderStyle	1-Fixed Single
Label# 1 Caption	Cd Player
Label# 1 Font size	18
Label# 1 Height	495
Label# 1 Left	1320
Label# 1 Top	480
Label# 2 Width	1935
Label# 2 Name	lblTrack
Label# 2 Alignment	1-Right Justify
Label# 2 Caption	Track
Label# 2 Font Style	Bold
Label# 2 Font size	12
Label# 2 Height	255
Label# 2 left	1200
Label# 2 Top	2280
Label# 2 Width	1215
Label# 3 Name	lblTrackNum
Label# 3 Caption	(blank)
Label# 3 Font stye	Bold
Label# 3 Font size	12
Label# 3 Height	375
Label# 3 Left	2520
Label# 3 Top	2280
Multimedia controll Name	mmcCD
Multimedia control Device Type	CDAudio

بعد از ایجاد فرم، کد زیر را در آن بنویسید :

```
1: Private Sub Form _ Load ()
2:   ‘Open the CD
3:   mmcCD.Command=‘‘Open’’
4: End Sub
5:
6: Private Sub Form _ Unload (Cancel As Integer)
7:      ‘Clean up the multimedia control when done
8:   mmcCD. Command=: Close”
9: End Sub
10:
11: Private Sub mmcCD _ Status Update()
12: ‘Update the track number in the lable
13: ‘lblTrack Num. Caption= mmcCD.Track
```

برنامه، ابزار CD را در خط ۳ باز می‌کند و وقتی برنامه بسته می‌شود دستور خط ۸ کنترل چند رسانه‌ای را از حافظه پاک می‌کند. خط ۱۴ هم شماره‌ی Track فعلی را در برچسب می‌نویسد. رویداد StatusUpdate با هر تغییری که در وضعیت ابزار رخ دهد، فعال می‌شود. علاوه بر این، بعد از سپری شدن زمان تنظیم شده در مشخصه‌ی UpdateInterval (که مقدار پیش فرض آن ۱۰۰۰ میلی ثانیه است) نیز این رویداد فعال خواهد شد.

شکل ۳ــ۲ برنامه‌ی پخش CD را نشان می‌دهد. این برنامه با وجود سادگی، کار می‌کند!



شکل ۳ــ۲ــ برنامه‌ی پخش CD بسیار ساده است.

۴ــ۱ــ۲ــ رفع خطا

کنترل چند رسانه‌ای نیز می‌تواند دچار مشکل شود و برنامه‌نویس باید برای چنین موقعیت‌هایی چاره‌اندیشی کند.

مشخصه‌ی Notify کنترل چند رسانه‌ای هرگاه که این کنترل فرمانی را انجام دهد، یک رویداد Done ایجاد می‌کند. خودتان هم می‌توانید بعد از پایان یک فرمان مشخصه‌ی Notify را با True مقداردهی کنید. اگر مشخصه‌ی Wait با True تنظیم شده باشد، تا زمانی که کنترل چند رسانه‌ای فرمانی را پایان نداده باشد، به برنامه اصلی برگردانده نخواهد شد. برای بررسی خطا می‌توانید از قطعه کد زیر استفاده کنید :

```
If (frmCD. mmCD. Error) then
        IntMsg= MsgBox (‘‘CD Not working propertly’’, vbCritical)
Else
        ‘Code continues here to play the CD Player
```

اگر مانند گذشته از دستور On Error Goto استفاده کنید، نمی‌توانید بدانید که کدام کنترل چند رسانه‌ای سبب بروز خطا شده است. با روش فوق بهتر می‌توانید خطاها را کنترل کنید.

کنترل چند رسانه‌ای مشخصه‌ای به نام Mode دارد که به وسیله‌ی آن می‌توان فهمید که کنترل در حال حاضر مشغول انجام چه کاری است. جدول ۳ــ۲ مقادیر این مشخصه را نشان داده است.

جدول ۳ــ۲ــ مقادیر مشخصه‌ی MODE کنترل چند رسانه‌ای

ثابت نام‌دار	مفهوم
mciModeNotOpen	کنترل چند رسانه‌ای هنوز باز نشده است.
mciModeStop	کنترل چند رسانه‌ای بسته شده است.
mciModePlay	کنترل چند رسانه‌ای در حال پخش است.
mciModeRecord	کنترل چندرسانه‌ای در حال ضبط است.
mciModeSeek	کنترل چندرسانه‌ای در حال جستجوی اطلاعات است.
mciModePause	کنترل چند رسانه‌ای به طور موقت متوقف شده است.
mciModeReady	کنترل چندرسانه‌ای آماده‌ی کار شده است.

مثال ۲ــ۲ــ برنامه‌ی پخش فایل‌های صوتی: صوت را به صورت فایل هم می‌توان در آورد. برای پخش این فایل‌ها به یک ابزار مرکب چند رسانه‌ای نیاز داریم. در این قسمت برنامه‌ای

برای پخش فایل‌های صوتی می‌نویسیم. برای نمایش اطلاعات فایل در حال پخش باید از مشخصه Mode استفاده کنیم. این برنامه فایل Windows\Media\Tada. wav\ را پخش خواهد کرد.

پروژه‌ی پخش CD را با نام جدیدی ذخیره کرده و سپس تغییرات زیر را در آن اعمال کنید :

۱ــ نام برچسب بالایی را به lbWav و عنوان آن را به نام Wave Player تغییر دهید. عرض آن را هم با ۲۴۱۵ تنظیم کنید.

۲ــ نام فرم را به frmWav تغییر داده و عنوان آن را Wave Music Player قرار دهید.

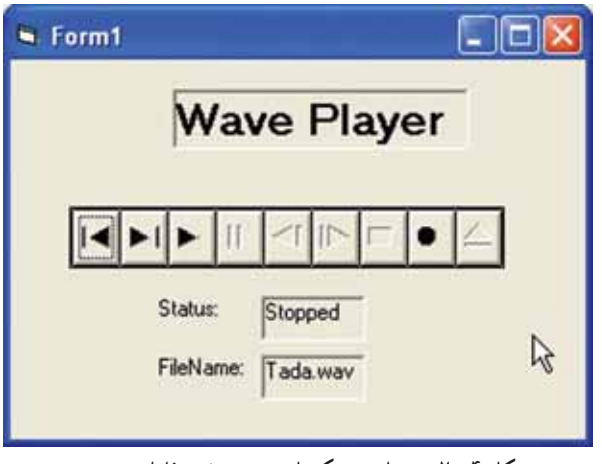
۳ــ نام کنترل چند رسانه‌ای را به mmcWav تغییر داده و مشخصه Device Type آن را با Wave Audio تنظیم کنید. نام فایل صوتی را (در مشخصه‌ی FileName) با Windows\Media\Tada.wav\ تنظیم کنید. برای سهولت از مشخصه‌ی Custom کنترل چند رسانه‌ای استفاده کنید.

۴ــ چون فایل صوتی اساساً دارای Track نیست، نام برچسب سمت چپ را به lblStatus و عنوان آن را به Status تغییر دهید.

۵ــ نام برچسب سمت راست را به lblStatus Value تنظیم کرده و آن را خالی رها کنید. عرض آن را با ۲۵۶۵ تنظیم کنید.

۶ ــ دو برچسب دیگر در زیر برچسب‌های فوق قرار داده و نام برچسب سمت چپ را lblFile و عنوان آن را : Filename قرار دهید. (دقت کنید که مشخصه‌های قلم برچسب‌های جدید با برچسب‌های قدیمی یکسان باشد.)

۷ــ نام برچسب سمت راست را lblFileValue گذاشته و آن را خالی رها کنید (شکل ۴ــ۲).



شکل ۴ــ۲ــ برنامه‌ی تکمیل شده‌ی پخش فایل صوتی

روال Form_Load() را به صورت زیر تغییر دهید :

```
Private Sub Form_ Load ()
        'Tell the Mulimedia control to open the WAVE player
        mmcWAV. Command= ''Open''
End Sub
```

تغییرات زیر را هم در روال Form _ unload بدهید :

```
Private Sub Form_Unload (Cancle As Integer)
                'Clean up the WAVE and form
                mmc WAV. Command = ''Close''
                Unload Me 'Unloads the form as well
End Sub
```

برنامه را اجرا کنید. روی دکمه‌ی پخش (►) کلیک کنید تا صوت پخش شود. برای پخش مجدد صوت ابتدا باید فایل را به عقب (|◄◄) برگردانید. اما برنامه هنوز کامل نیست : برچسب‌ها هنوز اطلاعات صحیح فایل را نمایش نمی‌دهند. کد زیر روال رویداد StatusUpdate() را نشان می‌دهد ؛ در این روال از مشخصه‌ی Mode استفاده شده است. MAV_ Status

```
1: Private Sub mciMAV_ Status Update ()
2: 'Display the status
3: If mmcWAV. Mode = mciModeNotOpen Then
4:    lblStatus Value(0). Caption = ''Not Ready''
5: Elself mmcWAV. Mode = mciMode Stop Then
6:    lblStatusValue(0). Caption=''Stopped''
7: ElseIf mmcWAV.Move= mciModePlay Then
8:     lblStatusValue(0). Caption = ''Play''
9: ElseIf mmcWAV. Mode = mciModeRecord Tehn
10:       lblStatusValue(0). Caption = ''Recoed''
11: ElselfmmcWAV.Mode = mciModePause Then
12:       lblStatus Value(0). Caption = ''Paused''
13: ElseIf mmc WAV. Mode = mciModeReady Then
14:       lblStatus Value(0). Caption = ''Ready''
15: End If
16: 'Display the filename being played
17: lblStatus Value (l). Caption = mmcWAV. FileName
18: End Sub
```

در این لیست فرض شده است که از آرایه کنترل استفاده کرده‌اید : اگر چنین نیست نام برچسب‌ها را در لیست تغییر دهید. برنامه را دوباره اجرا کنید. در اجرای این برنامه باید به چند نکته دقت کنید :

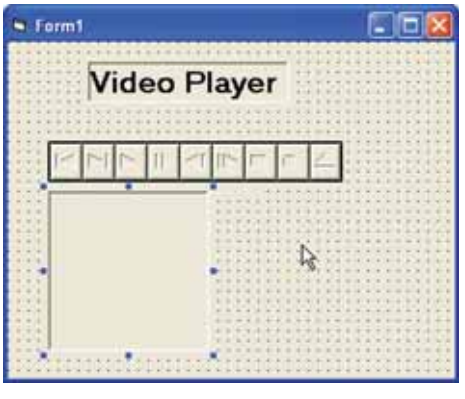
- چون فایل Tada.Wav کوچک است، عملاً امکان متوقف کردن آن وجود ندارد.
- پس از پخش فایل، باید دوباره آن را به عقب برگردانید.
- در این برنامه دکمه‌ی ضبط (●) فعال است و می‌توانید صوت موردنظر را به آخر فایل اضافه کنید.
- اگر می‌خواهید صوت را با نام دلخواه خود ضبط کنید، باید ویژگی‌های دیگری (از قبیل کادر محاوره‌ای ذخیره‌کردن فایل) به برنامه اضافه کنید.

۵ــ۱ــ۲ــ نمایش فایل‌های ویدیویی

نمایش فایل ویدیویی به همان سادگی پخش فایل‌های صوتی است. اما برای نمایش فایل ویدیویی به کمی کمک نیاز داریم : کمک از یک کنترل کادر تصویر.

برای نمایش فایل ویدیویی باید مشخصه‌ی DeviceType را با AVIVideo تنظیم کنید. مهم‌ترین نکته در پخش یک فایل ویدیویی آن است که کنترل چند رسانه‌ای خروجی ویدیو را به یک پنجره می‌فرستد نه به صفحه‌ی نمایش. برای تعیین پنجره‌ی خروجی فایل ویدیویی از مشخصه‌ی hWndDisplay کنترل چند رسانه‌ای استفاده می‌کنیم (در واقع هر کنترل یک پنجره مستقل است.) به این مشخصه نام کنترل کادر تصویر را می‌دهیم.

مثال ۳ــ۲ــ پروژه‌ی جدیدی باز کرده و یک کنترل چند رسانه‌ای روی آن قرار دهید. نام این کنترل را mmcVideo بگذارید. نام فرم را به frmVideo و عنوان آن را با Video Player تغییر دهید. یک کنترل کادر تصویر به نام picVide روی فرم قرار دهید (شکل ۵ــ۲).



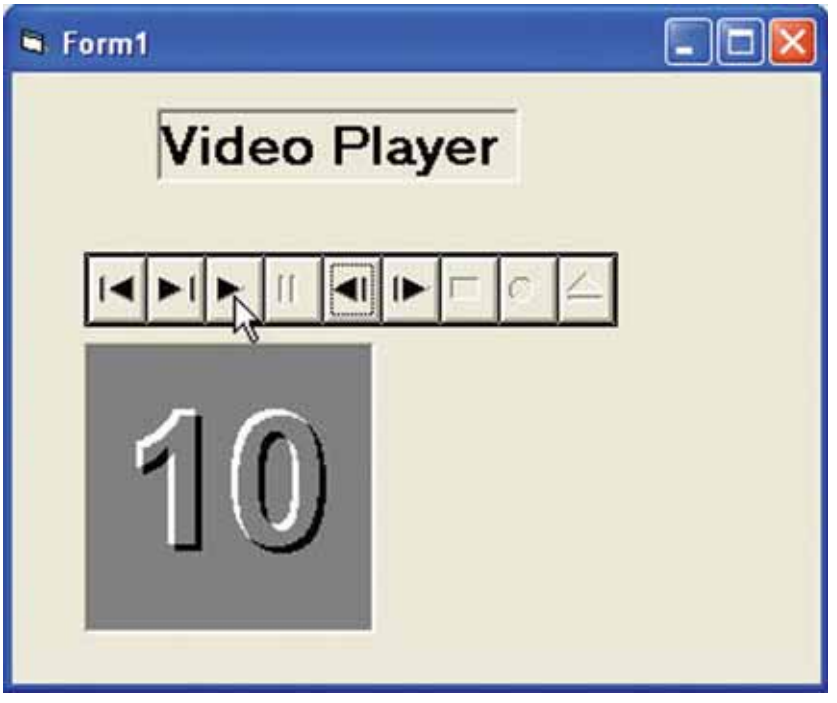
شکل ۵ــ۲ــ فایل ویدیویی در یک کادر تصویر نمایش داده خواهد شد.

مشخصه‌ی Devicetype کنترل mmcVideo را با AVIVideo مقداردهی کرده و از فایل Count24.Avi به عنوان فایل ویدیویی استفاده کنید. (این فایل را می‌توانید در پوشه‌ی Graphics\Videos\ محل نصب Visual Basic پیدا کنید.)

فقط کافی است کد زیر را به برنامه اضافه کنید تا یک برنامه‌ی پخش ویدیویی کامل داشته باشید.

```
1: Private Sub Form_ Load()
2:       'Open the video player
3:       mmcVideo.Command = "Open"
4:       'Connect the video player to the Picture Box
5:       mmcVideo. hWndDisplay= picVideo.hWnd
6: EndSub
```

برنامه را اجرا کنید ؛ شکل ۶ــ۲ اجرای برنامه‌ی پخش ویدیو را نشان می‌دهد.



شکل ۶ــ۲ــ برنامه‌ی پخش ویدیو بدون مشکل کار می‌کند!

مثال ۴ــ۲ــ در این پروژه‌ی عملی، برنامه‌ای می‌نویسیم که دارای ویژگی‌های زیر است :

- کادر محاوره‌ای About
- پخش یک فایل صوتی در هنگام نمایش کادر محاوره‌ای About
- یک پویا نمایی ساده با استفاده از کنترل کادر تصویر
- استفاده از Timer در پویانمایی
- آرایه‌ی کادر تصویر

پویانمایی این برنامه بسیار ساده و ابتدایی است ولی شما را با اصول حرکت‌دادن شیءها و پخش فایل‌های WAV آشنا خواهد کرد.

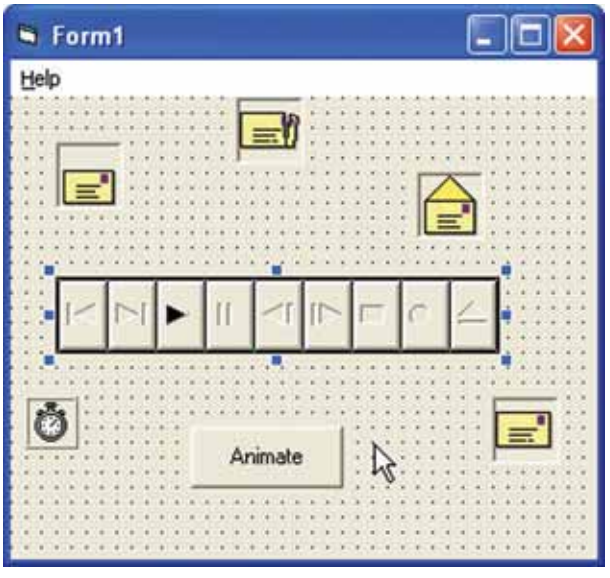
شکل ۷ــ۲ اجرای این برنامه را نشان داده است. در ابتدا پاکت بسته است و هیچ حرکتی در برنامه دیده نمی‌شود. اما هنگامی که روی دکمه‌ی Animation کلیک کنید، در پاکت باز شده و نامه‌ی داخل آن به پرو از درخواهد آمد. متعاقب آن عنوان دکمه به Stop تبدیل می‌شود و تا زمانی که کاربر این دکمه را کلیک نکند، پویانمایی متوقف نخواهد شد.

شکل ۷ــ۲ــ یک نامه در حال پرواز است.

شکل ۸ــ۲ کادر محاوره‌ای About برنامه را که بعد از انتخاب گزینه‌ی Help|About ظاهر خواهد شد، نشان می‌دهد.

شکل ۸ــ۲ــ در حین نمایش کادر محاوره‌ای About یک فایل WAV هم پخش خواهد شد.

ایجاد فرم اصلی برنامه: جدول ۴ــ۲ کنترل‌های برنامه و مشخصه‌های آن‌ها را نشان می‌دهد. در این برنامه از فایل‌های پوشه‌ی Graphics استفاده کرده‌ایم، بنابراین اگر این فایل‌ها را نصب نکرده‌اید، باید این کار را انجام دهید (یعنی باید دوباره برنامه‌ی نصب Visual Basic را از روی CD آن اجرا کنید). کنترل چند رسانه‌ای را هم به جعبه ابزار خود اضافه کنید. چون تعدادی از کنترل‌های این برنامه نامریی هستند، مکان آن‌ها روی فرم اهمیت چندانی ندارد. مکان کنترل کادر تصویر هم چندان مهم نیست چون در طول برنامه تغییر خواهد کرد. شکل ۹ــ۲ فرم برنامه را در مرحله‌ی طراحی نشان می‌دهد.



شکل ۹ــ۲ــ فرم تکمیل‌شده‌ی برنامه‌ی پویانمایی

جدول ۴ــ۲ــ مشخصه‌های کنترل‌ها

کنترل	مقدار مشخصه
Form Name	Frm Envelope
Form Caption	Mailing a Letter
Form Height	5790
Form Width	7845
Menu option #1 Name	mnuHelp
Menu option #1 Caption	& Help
Menu option #2 Name	mnuHelpabout
Menu option #2 Caption	& About
Command button Name	cmdAni
Command button Caption	& Animate
Command button Left	2940
Command button Top	2880
Timer Name	tmrAni
Timer Enabled	False
Timer Interval	300
Timer Left	1410
Timer Top	3405
Picture box# 1 Name	picAni 1
Picture box# 1 Height	495
Picture box# 1 Left	3330
Picture box# 1 Picture	Common\Graphics\Icons\Mail\Mai101
Picture box# 1 Top	1485
Picture box# 1 Width	1215
Picture box# 2 Name	picAni2(0)
Picture box# 2 Height	495
Picture box# 2 Left	5895
Picture box# 2 Picture	Common\Graphics\Icons\Mail\Mai101
Picture box# 2 Top	2520
Picture box# 2 Width	1215
Picture box# 3 Name	picAni2(1)
Picture box# 3 Height	495
Picture box# 3 Left	5520
Picture box# 3 Picture	Common\Graphics\Icons\Mail\Mai101
Picture box# 3 Top	3240
Picture box# 3 Visible	False
Picture box# 1 Width	1215
Picture box# 4 Name	picAni2(1)
Picture box# 4 Height	495
Picture box# 4 Left	3960
Picture box# 4 Picture	Common\Graphics\Icons\Mail\Mai103
Picture box# 4 Top	1080
Picture box# 4 Visible	false
Picture box# 4 Width	1215
Multimedia control Name	mmcEnc
Multimedia control Device Type	WaveAudio
Multimedia control PlayEnabled	True
Multimedia control Filename	\Windows\Media\Chimes.wav
Multimedia control Left	2520
Multimedia control Top	4080
Multimedia control Visible	False
Multimedia control Width	3540

کد زیر را در برنامه وارد کنید. پویانمایی ساده‌ی این برنامه حاصل سه تصویر است.

```
1: Private Sub cmdAni_Click()
2:          'Use the command button to control the animation
3:          If cmdAni.Caption = ''& Animate'' Then
4:          cmdAni.Caption = ''& Stop''
5:          tmrAni.Enabled = True
6: Else
7:          cmdAni.Caption = ''& Animate''
8:          tmrAni.Enabled = False
9: End If
10: End Sub
11:
12: Private Sub mnuHelpAbout_Click()
13:          mmcEnv.Command = ''Open''
14:          mmcEnv.Command = ''Play''
15:          FrmAbout.Show
16: End Sub
17:
18: Private Sub tmrAni_Timer
19:          'Determine the correct picture
20:          'location to display
21:          '
22:          'The following variable begins at zero
23:          'and retains its value every time the
24:          'procedure executes.
25:          Static intCounter As Integer
26:
27:          Select Case intCounter
28:               Case 0:
29:                    picAni 1. Picture = pic Ani2(1). Picture
30:                    picAni2 (2).Visible = True
31:                    picAni2 (2).Left = 3840
32:                    picAini2(2).Top=1220
33:                    intCounter =1
34:               Case l:
35:                    picAnil.Picture = picAni2(1). Picture
```

```
36:                 picAni2(2).Visible = True
37:                 picAni2(2).Left = 4040
38:                 picAni2(2). Top = 1120
39:                 intCounter = 2
40:             Case 2:
41:                 picAnil. Picture = picAni2(1). Picture
42:                 picAni2(2).Visible = True
43:                 picAni2(2).Left = 4240
44:                 picAni2(2).Top = 1220
45:                 intCounter = 3
46:             Case 3:
47:                 picAni l.Picture = picAni2(0). Picture
48:                 picAni2(2). Left= 4440
49:                 picAni2(2).Top = 1320
50:                 intCounter = 4
51:             Case 4:
52:                 'Stop the animation
53:                 picAni1.Visible=true
54:                 intCounter = 0
55:                 picAni2(2).Visible = False
56:             End Select
57: End Sub
```

روال رویداد cmdAni_Click() عنوان دکمه‌ی انیمیشن را بین دو حالت (Animation و Stop) تغییر می‌دهد و بسته به هر حالت، Timer را فعال (Enable) و غیرفعال (Disable) می‌کند. با غیرفعال کردن Timer، پویانمایی برنامه هم متوقف خواهد شد چون روال رویداد tmrAni_Timer() (که مسئول اجرای انیمیشن است) فقط در حالتی که مشخصه Timer Enabled مقدار True داشته باشد، اجرا خواهد شد.

پویانمایی برنامه در روال رویداد tmrAni_Timer()، که هر ۳۰۰ میلی ثانیه تکرار می‌شود، پیاده‌سازی شده است. در این روال یک دستور Select Case وجود دارد (خط ۲۷) که کار خود را براساس مقدار یک متغیر استاتیک به نام intCounter (که در خط ۲۵ تعریف شده است) انجام می‌دهد. مقدار این متغیر می‌تواند از ۰ تا ۴ تغییر کند.

هر دستور Case چهار کار انجام می‌دهد :

- به جای آیکون پاکت بسته، آیکون پاکت باز را قرار می‌دهد (خطوط ۲۹ و ۳۵).

- آیکون را نمایش می‌دهد (خطوط ۳۰ و ۳۶) .
- با تغییر مشخصه‌های Left و Top (خطوط ۳۱ و ۳۲)، آیکون را حرکت می‌دهد.
- متغیر intCounter را افزایش می‌دهد تا دفعه‌ی بعد Case دیگری اجرا شود.

هنگامی که مقدار intCounter به ۴ رسید (آخرین دستور Case)، مقدار آن مجدداً با ۰ مقداردهی شده و همه چیز از سر گرفته می‌شود (البته مشروط به اینکه Timer همچنان فعال باشد).

روال mnuHelpAbout_Click() کادر محاوره‌ای About را نمایش داده و فایل صوتی Chimes. Wav را پخش خواهد کرد. چون برای این کادر محاوره‌ای از الگوی فرم About استفاده کرده‌ایم، نیازی به کدنویسی ندارد.

ایجاد کادر محاوره‌ای About

در پنجره‌ی پروژه کلیک راست کرده و با انتخاب Add|Form یک فرم About به برنامه اضافه کنید. جدول ۵ــ۲ مشخصه‌های کنترل‌های این فرم را نشان می‌دهد.

جدول ۵ــ۲ــ مشخصه‌های کنترل‌های فرم About

کنترل	مقدار مشخصه
LblDescription Caption	Seeing Simple animation and hearing a sound
LblDescription Font size	14
LblDescription Font style	Bold

خودآزمایی و تحقیق

۱ــ چرا Visual Basic تمام دکمه‌های کنترل مالتی مدیا را فعال نمی‌کند؟

۲ــ خاصیت Mode چه کاری انجام می‌دهد؟

۳ــ چگونه می‌توان تعداد رویدادهای StatusUpdate در واحد زمان را افزایش داد؟

۴ــ چرا برای نمایش فایل‌های ویدیویی به کنترلی مانند جعبه تصویر هم نیاز داریم؟

۵ــ کنترل چند رسانه‌ای چگونه می‌فهمد که خروجی ویدیو را باید به کدام کادر تصویر بفرستد؟

۶ــ به برنامه‌ی پخش فایل صوتی یک کادر محاوره‌ای باز کردن فایل اضافه کنید تا کاربر بتواند فایل مورد نظرش را باز و پخش کند.

۷ــ یک آلبوم صوتی ساده ایجاد کنید.

فصل سوم

API[1] ویندوز

هدف‌های رفتاری: فراگیر پس از پایان این فصل، خواهد توانست :

۱ــ مفهوم API ویندوز را شرح دهد.

۲ــ دلیل نیاز برنامه‌های شما به روال‌های ویندوز را بیان کنید.

۳ــ کتابخانه‌های پیوند دینامیکی (فایل‌های DLL) را شرح دهد.

۴ــ نحوه‌ی اتصال ویژوال بیسیک به روال‌های API ویندوز از طریق دستور Declare را توضیح دهد.

در این فصل، نحوه‌ی دسترسی به روال‌های داخلی ویندوز شرح داده می‌شود. هرچند ویژوال بیسیک می‌تواند تقریباً همه کارهای مورد نیاز شما را انجام دهد، ولی در برخی از برنامه‌ها اگر بخواهید از ویژوال بیسیک برای انجام بعضی قابلیت‌ها استفاده کنید، برنامه‌نویسی مشکل می‌شود. خوشبختانه در چنین شرایطی روال‌هایی در ویندوز وجود دارند که درون فایل‌های DLL ذخیره شده‌اند.

با استفاده از روال‌های ویندوز می‌توانید قدرت ویژوال بیسیک را در برنامه‌های خودتان افزایش دهید و از آن بخواهید کارهایی را انجام دهد که فقط ویندوز حق اجرای آن‌ها را دارد. در این فصل، نه تنها نحوه‌ی دسترسی به این روال‌های ویندوز توضیح داده می‌شود بلکه تعدادی از این روال‌ها که می‌توانید با آن‌ها کار کنید، بررسی می‌شوند.

۱ــ۳ــ API ویندوز

API ویندوز مجموعه‌ای از روال‌هاست که در دسترس برنامه‌نویس ویژوال بیسیک می‌باشد.

۱ــ Application Programming Interface

به عبارت دیگر، این روال‌های API دقیقاً مانند توابع داخلی خود ویژوال بیسیک عمل می‌کنند. وقتی لازم است تا از کد یک روال API استفاده کنید، برنامه ویژوال بیسیک آن روال را فراخوانی می‌کند. زمانی که API ویندوز به پایان رسید، کنترل به برنامه برمی‌گردد و اجرای آن ادامه پیدا می‌کند.

API سرنام Application Programming Interface به معنای رابط برنامه‌نویسی کاربردی است و به مجموعه‌ای از روال‌های داخلی ویندوز اطلاق می‌شود که می‌توانید آن‌ها را از ویژوال بیسیک فراخوانی کنید.

تمام روال‌های API ویندوز در فایل‌های خاصی با نام DLL ذخیره می‌شوند. چند هزار روال API وجود دارد که می‌توانید آن‌ها را به کار ببرید. این روال‌های API درون فایل‌هایی وجود دارند که در پوشه‌های Windows و Windows\System ذخیره شده‌اند. هنگام نصب ویندوز، فایل‌های DLL، هم نصب می‌شوند. بنابراین، به‌طور خودکار به این کتابخانه‌ها دسترسی دارید.

DLL سرنام Dynamic Link Library به معنای کتابخانه پیوند پویاست. فایل‌های DLL در دسترس برنامه‌هایی که به زبان ویژوال بیسیک و زبان‌های دیگری (که از DLL پشتیبانی می‌کنند) نوشته شده‌اند، قرار دارد.

اکثر فایل‌های DLL دارای پسوند DLL. یا EXE. هستند. هر برنامه‌ای که می‌نویسید، به فایل‌های DLL ویندوز دسترسی دارد.

سه فایل DLL که معمولاً به کار می‌روند، عبارتند از :

- **USER32.DLL** : شامل توابعی است که محیط ویندوز و رابط کاربر مثل مکان‌نماها، منوها و پنجره‌ها را کنترل می‌کنند.
- **GDI32. DLL** : شامل توابعی است که خروجی برنامه به صفحه نمایش و ابزارهای دیگر را کنترل می‌کنند.
- **KERNEL32. DLL** : شامل توابعی است که سخت‌افزار و رابط‌نرم افزار داخلی ویندوز را کنترل می‌کنند. اکثر روال‌های مربوط به حافظه، فایل و دایرکتوری درون KERNEL32.DLL قرار دارند.

این سه فایل، اغلب روال‌ها یا توابع API را نگه می‌دارند. شما می‌توانید این توابع را در برنامه‌های ویژوال بیسیک خودتان فراخوانی کنید. با یک نگاه کوتاه به پوشه‌های Windows و Windows\System چند کتابخانه پیوند دینامیکی دیگر را نیز می‌بینید مثل COMDLG.DLL، MAPI32.DLL، NETAPI32.DLL و WINMM.DLL. هم‌چنان‌که مایکروسافت قابلیت‌هایی

را به سیستم عامل اضافه می‌کند، فایل‌های جدید DLL هم ظاهر می‌شوند.

هنگامی که برنامه‌ی جدیدی را در سیستم خودتان نصب می‌کنید، این برنامه DLL خاص خودش را ارایه می‌کند. بنابراین، در طول زمان تعداد زیادی فایل DLL روی سیستم خودتان خواهید داشت.

۲ــ۳ــ فایل‌های DLL

اصطلاح پیوند پویا معنای خاصی برای برنامه‌نویسان دارد. وقتی گفته می‌شود که یک روال با یک برنامه پیوند دینامیکی دارد، بدین معناست که این روال (سابروتین یا تابع) تا قبل از کامپایل برنامه، به آن متصل نمی‌شود. این تابع فقط در زمان اجرا در دسترس می‌باشد. توابعی که درون پنجره کد می‌نویسید دارای پیوند استاتیکی می‌باشند یعنی هر وقت برنامه را کامپایل می‌کنید، این توابع با بقیه کد اصلی ترکیب می‌شوند. اما فایل‌های DLL با برنامه ترکیب نمی‌شوند. برنامه‌ی شما به این روال‌ها در زمان اجرا دسترسی دارد اما فایل EXE برنامه شامل روال‌های واقعی DLL نمی‌باشد.

> **نکته:** مزیت استفاده از روال‌های DLL آن است که چند برنامه اجرایی تحت ویندوز می‌توانند به یک روال از فایل DLL دسترسی داشته باشند. علاوه بر این همه کاربران باید روال‌های استاندارد DLL را داشته باشند. از آن‌جایی که برای اجرای یک برنامه ویژوال بیسیک، وجود ویندوز ضروری است، پس فایل‌های DLL در دسترس می‌باشند.

۳ــ۳ــ دستور Declare

برای فراخوانی روال‌های API ویندوز باید از یک دستور خاص با نام Declare استفاده کنید. در مورد توابع داخلی ویژوال بیسیک به دستور Declare نیاز ندارید چون ویژوال بیسیک طرز کار توابع خودش و آرگومان‌های این توابع را می‌شناسد. اما روال‌های API خارج از میدان دید ویژوال بیسیک می‌باشند. بنابراین باید از این دستور استفاده کنید.

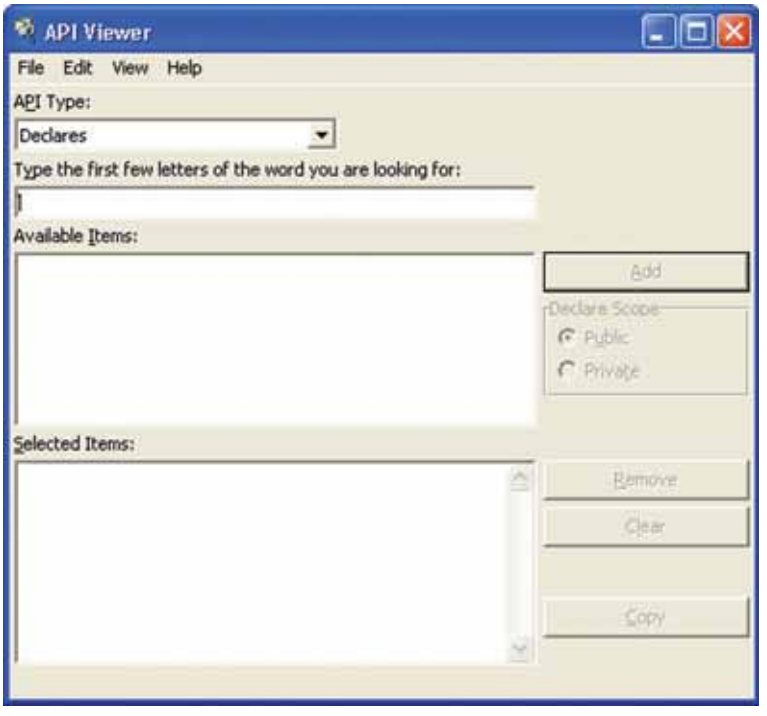
۱ــ۳ــ۳ــ ابزار API Viewer

ویندوز شامل هزاران روال API است که می‌توانید آن‌ها را فراخوانی کنید. حتی دانستن قالب تعداد کمی از این روال‌ها هم کار مشکلی است. در این رابطه، ویژوال بیسیک یک ابزار خاص به نام API Viewer دارد که برای راهنمایی در مورد قالب روال‌های API می‌توانید از این ابزار استفاده کنید.

ابزار API Viewer، روال‌های API را نمایش می‌دهد و آن‌ها را از نظر موضوع دسته‌بندی می‌کند طوری که می‌توانید روال‌های مورد نیاز خود را به سادگی پیدا کنید.

دکمه‌ی Copy روی API Viewer اطلاعات مربوط به اعلان انتخاب شده را درون Clipboard ویندوز کپی می‌کند. همچنین اگر قبل از کلیک‌کردن روی دکمه‌ی Copy، گزینه‌های Public یا Private این ابزار را کلیک کنید دیگر مجبور نیستید شناسه‌های اعلان را به صورت دستی تغییر دهید. زیرا API Viewer کلمات کلیدی مناسب را در دستور Declare مشخص می‌کند.

برای دسترسی به ابزار API Text Viewer ، از Start گزینه Programs، سپس گزینه Microsoft Visual Studio 6.0 و سپس گزینه‌ی Microsoft Visual Studio 6.0 Tools و در پایان API Text Viewer را انتخاب کنید.

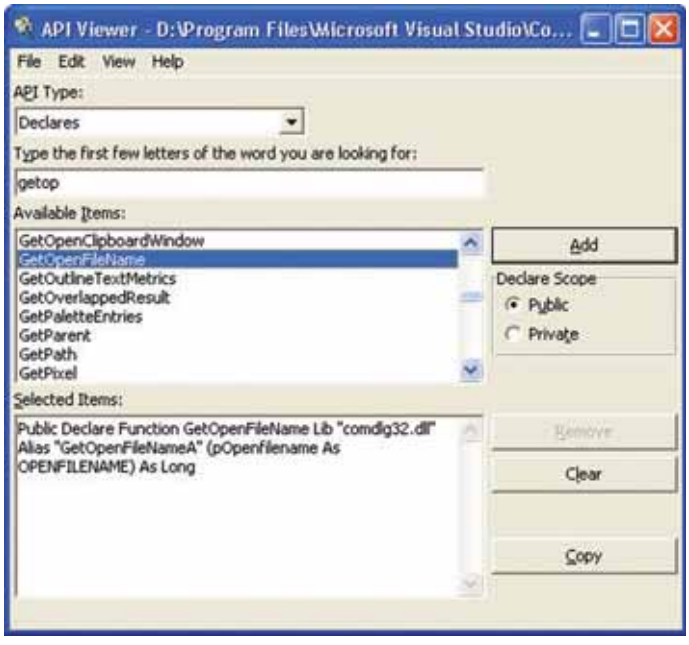


شکل ۱ــ۳ــ می‌توانید از طریق APIViewer به سادگی قالب روال‌های API را تعیین کنید.

API Viewer اطلاعات اساسی خود را از فایل‌های متنی APILOAD.txt، MAPI32.txt و WIN32API. txt پیدا می‌کند. این فایل‌ها همراه API Viewer روی سیستم نصب می‌شوند.

از آن‌جایی که اکثر روال‌های API که موردنظر شما هستند، در فایل WIN32API.txt قرار

دارند. گزینه Load Text File را از منوی File پنجره API Viewer انتخاب و سپس فایل WIN32API.txt را انتخاب کنید.



شکل ۲ــ۳

دقت کنید کادر لیست موجود در بالای پنجره دارای عنوان API Type است. با بازکردن این کادرلیست، سه مقدار زیر را مشاهده می‌کنید :

- **Constants** : همه ثابت‌های نامگذاری شده که فایل API بارگذاری شده تشخیص می‌دهد را فهرست می‌کند.
- **Declares** : همه اعلان‌هایی که درون فایل API بارگذاری شده ظاهر می‌شوند را فهرست می‌کند.
- **Types** : همه انواع داده‌ها که فایل API بارگذاری شده تشخیص می‌دهد را فهرست می‌کند.

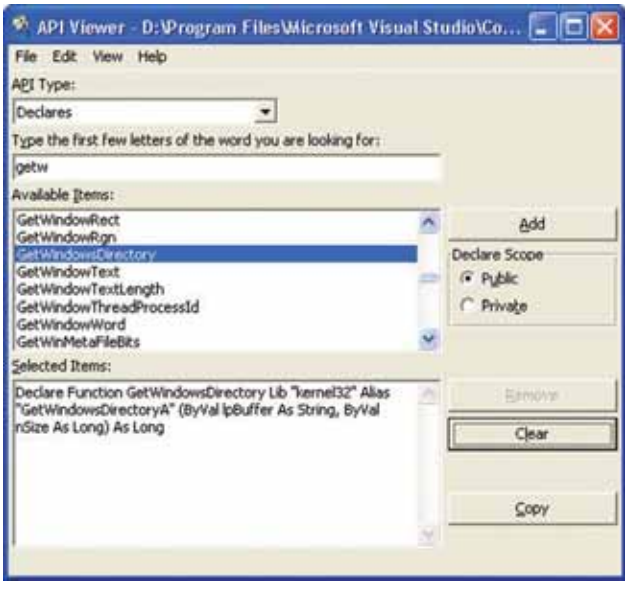
کادر لیست Available Items شامل کلیه روال‌های API ویندوز (مربوط به فایلی که بارگذاری کرده‌اید) و نوع مقداری است که می‌خواهید مشاهده کنید. مثلاً اگر می‌خواهید دستور Declare مورد نیاز روال GetWindowsDirectory را پیدا کنید، مراحل زیر را دنبال نمایید :

۱ــ از لیست API Type گزینه Declares را انتخاب کنید. چند دستور Declare درون لیست Available Items ظاهر می‌شوند.

۲ــ می‌توانید چند حرف اول یک دستور Declare خاص را درون کادر متن تایپ کنید تا این دستور به سرعت پیدا شود. بدین منظور Getw را تایپ کنید. همه گزینه‌هایی که با این حروف شروع می‌شوند درون کادر لیست Available Items ظاهر می‌شوند.

۳ ــ این فهرست را مرور کنید تا گزینه GetWindowsDirectory را پیدا کنید.

۴ــ روی گزینه GetWindowsDirectory دابل کلیک کنید تا دستور Declare مورد نیاز تابع مطابق شکل ۳ــ۳ نمایش داده شود.



شکل ۳ــ۳ــ API Viewer دستور Declare مورد نیاز برای دستوری که انتخاب کرده‌اید را نمایش می‌دهد.

اکنون می‌توانید تمام دستور Declare را کپی کنید و آن را درون پنجره کد برنامه خودتان قرار دهید.

۴ــ۳ــ تعریف تابع API

قبل از این‌که کار با توابع را شروع کنیم بهتر است با فرم کلی توابع API آشنا شویم :

[Public| Private] Declare Function Function - name Lib "DllFilename"-

[alias "Function alias''] (argument _ List) as data-type

دستور Declare برای تعریف تابع استفاده می‌شود. این دستور می‌تواند داخل یک ماژول یا فرم به کار رود. اگر داخل فرم استفاده شود کلید واژه Private برای آن به‌کار رفته شود و اگر داخل ماجول به کار رود، می‌تواند با یکی از کلید واژه‌های Public یا Private استفاده شود. در مورد توابع داخلی ویژوال بیسیک به دستور Declare نیاز ندارید، چون ویژوال بیسیک طرز کار توابع خودش و آرگومان‌های این تابع را می‌شناسد. اما روال‌های API خارج از میدان دید ویژوال بیسیک می‌باشند. بنابراین باید از این دستور استفاده کنید.

Function name، نام تابع API است.

Dll-Filename، نام فایل Dllای است که این تابع داخل آن قرار دارد. این نام شامل مسیر نباید باشد زیرا مسیر این فایل‌ها ثابت است. نوشتن پسوند این فایل الزامی نیست.

Function alias، این پارامتر اختیاری است. نامی است که داخل فایل dll قرار دارد.

Argument List، لیستی از آرگومان‌های تابع می‌باشد. این لیست نشان می‌دهد چه تعداد متغیر و از چه نوعی باید به تابع فرستاده شود. نحوه تعریف آرگومان‌ها به شکل زیر است :

[byval byRef] argument - Name as data - type

که در آن argument - Name نام آرگومان مورد نیاز است. انتخاب این نام اختیاری است و - data type نوع آرگومان را مشخص می‌کند.

کلید واژه byval، byRef روش‌های خاص ارسال پارامتر به تابع API است. این دو روش کاملاً متمایز هستند. روش و متد byRef به طور معمول استفاده می‌شود مگر اینکه به صراحت از byval استفاده شود. به همین دلیل شما در موقع تعریف API کلمه byRef را نمی‌بینید و فقط به byval نیاز است.

byvalue به این مفهوم است که تابع API نتواند مقدار این متغیر را تغییر دهد. ولی byRef به معنی «با مرجع» می‌باشد (by Refrence) و به این مفهوم است که تابع API می‌تواند مقدار این متغیر را تغییر دهد. با این روش نمی‌توان یک مقدار ثابت را ارسال کرد.

Data -type نوع داده‌ای که تابع برمی‌گرداند.

۱ــ۴ــ۳ــ تابع Messagebeep

یکی از ساده‌ترین روال‌های API است. این تابع یکی از دو کار زیر را انجام می‌دهد :

اگر آرگومان ارسالی به این تابع مثبت باشد، صدای بوق از طریق کارت صوتی کامپیوتر به گوش

می‌رسد.

اگر آرگومان ارسالی به این تابع منفی باشد، صدای بوق از طریق بلندگوی کامپیوتر به گوش می‌رسد.

شکل کلی این تابع به صورت زیر است :

```
Private Declare Function messagebeefLib 'user32' alias 'message - beep' (byval
wtype as long) as long
```

دستور Declare دقیقاً به ویژوال بیسیک اعلام می‌کند که چگونه تابع messagebeep را پیدا کند و چگونه مقادیر را منتقل نماید. این تابع درون فایل user32.dll است.

مثال ۱ــ۳ــ شنیدن صدای بوق: در وسط فرم دکمه‌ی Cmdbeep قرار دهید. می‌خواهیم با کلیک روی آن صدای بوق را بشنویم.

ابتدا API Viewer را اجرا کنید. فایل win32API.txt را بارگذاری کنید و APIType را به Declare تغییر دهید. فهرستی از روال‌های API درون AvailableItem ظاهر می‌شود. روی گزینه Messagebeep دابل کلیک کنید تا اعلان این روال درون جعبه Selected Item نمایش داده شود. سپس روی گزینه copy کلیک کنید. به صفحه کدنویسی برگشته و آن را در ناحیه general declaration، Paste کنید.

```
Private Declare Function messagebeep Lib 'user 32' alias 'message beep'-
(byval wtype as long) as long
Private sub cmdbeep _ clicki()
Dim Beeper as Variant
Beeper = messagebeep (1)
End sub
```

اگر با کلیک روی دکمه صدایی نشنیدید آرگومان را به ۱- تغییر دهید تا صدا را از بلندگوی کامپیوتر بشنوید. حتی اگر کارت صوتی نداشته باشید یا بلندگوی کامپیوتر خاموش باشد، باز هم بلندگوی داخلی کامپیوتر صدایی تولید می‌کند.

۲ــ۴ــ۳ــ تابع Gettick count

این تابع مدت زمانی که کامپیوتر روشن است را با واحد میلی‌ثانیه نشان می‌دهد. بنابراین برای تبدیل این واحد به ثانیه بایستی آن را به هزار تقسیم کنیم.

شکل کلی این تابع به صورت زیر است :

```
Private Declare Function Get Tickcount Lib “Kene132” AliasGet Tickcount
() as Lang
```

۳ــ۴ــ۳ــ تابع Sndplaysound

این تابع یک فایل صوتی از نوع wave را پخش می‌کند.

شکل کلی این تابع به صورت زیر است :

```
Private Declare Function sndplaysound Lib 'winmm. dll”
        alias ‘‘sndplaysoun’’ (byval Lpazsound Name as
                string, byval uflags as long) as long
```

آرگومان lpszSoundName نام فایلی صوتی است که قرار است پخش شود.

آرگومان uflags مقدار ثابتی است که نحوه پخش فایل را مشخص می‌کند.

این تابع درون فایل winmm.dll قرار دارد.

> **نکته:** توابع API مقادیر ثابت دارند که می‌توان از پنجره API viewer، با انتخاب Constants به جای Declare ثابت موردنظر را انتخاب کرده و مانند Declare به صفحه کدنویسی بیاورید.

مثال ۲ــ۳ ــ می‌خواهیم با کلیک روی فرم فایل صوتی tada.war پخش شود.

```
Private Declare Function sndplaysound
 lib "winmm.dll” alias - "sndplaysound”
(byval Lpszsound Name as string, byval uflags as long) as long
Private sub Form-click()
Dim a as long
    A=sndplaySound("windows\media\tada.wav”), l)
    End Sub
```

مثال ۳ــ۳ ــ این مثال به وسیله شیء CommonDialog یک فایل صوتی از نوع WAVE را انتخاب و به وسیله تابع SndplaySound آن را پخش می‌کند.

```
Dim a as long
Private Declare Function sndplay sound .....
```

```
Private sub form - bad
CDBox. Filter = "wave file (*.wav) * .wav" endsvb
Private sub commandi-click
A = sndplaysound (Cdbox. filename, 1)
End sub
```

۴ــ۴ــ۳ــ تابع Getlocaltime

ساعت و تاریخ سیستم را با جزییات آن در اختیار برنامه قرار می‌دهد.

شکل کلی این تابع به صورت زیر است:

```
Private Declare getlocaltime Lib
kernel32(Lpsystemtime as system time)
```

● ***آرگومان Lpsystemtime:*** رکوردی است که فیلدهای آن اجزای تاریخ و ساعت هستند.

این تابع درون فایل kernel32 قرار دارد.

مثال ۴ــ۳ــ فرم زیرتاریخ و ساعت روز را به تفکیک روز ــ ماه ــ سال و ساعت ــ دقیقه ــ ثانیه ــ صدم ثانیه می‌دهد.

```
Private Declare getlocaltime Lib kernel32(Lpsystemtime as system time)
Private Type systemtime
Myear as integer
Mmonth as integer
Mday as integer
Mhour as integer
Mminute as integer
Msecond as integer
End Type
Mmillisecond as integer
Private sub Form_click ( )
Dim mytime as systemtime
Me.autoRedraw = true
Getlocaltime mytime
Me. Print the local Date is: &mytime. mmonth& - mytime. mDay-&-& mytime.myear
Me.Print The local time is: & mytime. mhour&:&mytime.mminute
&:&mytime. msecond&:&mytime.mmillisecond
end sub
```

۵ــ۴ــ۳ــ تابع getdrivetype

نام درایو را می‌پذیرد و نوع آن را اعلام می‌کند.

شکل کلی آن به صورت زیر است :

```
Private Declare Function getdrivetype Lib “kernel32”alias
getdrive type (byval ndrive as string) as long
```

* آرگومان ndrive نام درایو موردنظر است.

این تابع درون فایل Kernel32.dll قرار دارد.

۶ــ۴ــ۳ــ تابع Swapmousebutton:

عملکرد کلیدهای چپ و راست ماوس را عوض می‌کند.

شکل کلی این تابع به صورت زیر است :

```
Private Declare Function swapmousebutton Lip ‘‘user32’’ (byval bswap as long)
as long
```

آرگومان bswap مقدار True یا false می‌پذیرد. اگر True باشد کلیدها عوض می‌شوند و اگر False باشد کلیدها عادی هستند.

این تابع درون فایل user 32.dll قرار دارد.

۷ــ۴ ــ۳ــ تابع lschild

بررسی می‌کند فرم جاری فرزند فرم MDI است.

شکل کلی این تابع به صورت زیر است :

```
Private DeclareFunction lsxhild Lib "user32’’ (byval hwndparent as long, byval
hwnd as long) as long
```

آرگومان hwndparent : هندل فرم والد یا MDI

آرگومان Hwnd : هندل فرم مورد بررسی

این تابع درون فایل user32.dll قرار دارد.

مثال ۵ ــ۳ــ فرم MDIای داریم به همراه فرمی دیگر می‌خواهیم بدانیم این فرم، فرزند MDI هست یا خیر؟

```
Private DeclareFunction Ischild Lib "user32’’(byval
hwndparent as long, byval hwnd as long) as long
private sub form_loadil)
if Ischild (MDIform1. hwnd, me.hwnd) = 1 Then
```

```
Msgbox "this form is a child", Vbinformation+ vbokonly,App.title else
Msgbox "this form is not a child" , Vbinformation + vbokonly,App. title
 endif
End sub
```

۸ـ۴ـ۳ـ تابع ExitwindowsEx

با این تابع می‌توان کامپیوتر را خاموش کرد یا آن را دوباره راه‌اندازی یا Logoff کرد.

شکل کلی این تابع به صورت زیر است :

```
Private Declare Function Exitwindows Ex Lib User32Alias "ExitwindowsEx" (Byval
uflags As long, Byval dwReserred As long) As
```

- آرگومان Uflags : شامل مقادیر ثابت زیر می‌تواند باشد :

EWX-logoff سبب می‌شود کامپیوتر logoff شود.

EWX-Shutdown سبب می‌شود کامپیوتر خاموش شود.

EWX-Reboot سبب می‌شود کامپیوتر دوباره راه‌اندازی شود.

EWX-Force مقادیر بالا بایستی با این مقدار OR شود.

- آرگومان dwreserved : در مقدار False یا true را می‌پذیرد. که true عمل خواسته شده را انجام می‌دهد False عمل خواسته شده را انجام نمی‌دهد.

این تابع درون فایل User32 قرار دارد.

مثال ۶ـ۳ـ برنامه زیر دستگاه را دوباره راه‌اندازی می‌کند.

```
Private Declare Function Exitwindows EX Lib "User32" "ExitwindowsEx" (Byval
uflags As long, Byval dwResered As long) As
Const EWX-Shutdown = 1
Const EWX-REBOOT = 2
Const EWX-FORCE = 4
Private Sub Form-Click ()
Msg = Msgbox "this program is going to Reboot your computer - Press ok to
Continue or cancel to stop". , Vbcritical + vbokcancel)
if msg = vbcancel Then End
Ret & = ExitwindowsEx (EWX-Force or EWX-REBOOT, )
End sub
```

۹ــ۴ــ۳ــ تابع Shoescrollbar

نوار اسکرول افقی یا عمودی را به فرم یا کنترلی می‌دهد.

شکل کلی این تابع به صورت زیر است :

Private Declare Function Showscrollbar Lib User32 (byval hwnd) as Long, byval byval as long) as long

- آرگومان hwnd : هندل کنترل موردنظر را می‌گیرد.
- آرگومان wbar : می‌تواند شامل مقادیر ثابت زیر باشد :

SB-HORZ : نوار اسکرول افقی

SB-VERT : نوار اسکرول عمودی

SB-BOTH : هر دو نوار اسکرول

- آرگومان bshow : در مقدار False نوار اسکرول دیده نمی‌شود.

این تابع درون فایل user32 قرار دارد.

مثال ۷ــ۳ــ برای فرم دو نوار اسکرول افقی و عمودی قرار می‌دهد.

```
Private Declare Function Showscrollbar "Lib User32" (byval hwnd as long, byval
wbar As long,byval bshow as long) as long
Const SB-HORZ= ∅
Const SE-VERT = 1
Const SB-BOTH = 3
Private Sub Form-load()
ShowSorollbar(Form1. Hwnd, SB-BOTH , True)
End sub
```

خودآزمایی و تحقیق

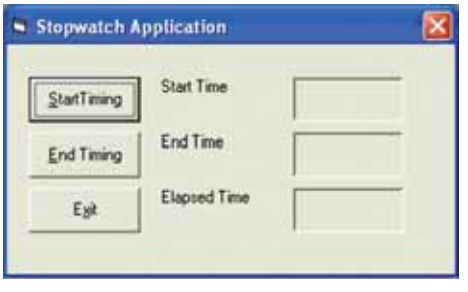
(توضیح: تمام تمرین‌های این فصل را با استفاده از توابع API بنویسید.)

۱ــ یکی از ساده‌ترین روال‌های API تابع ( )MessageBeep است. این تابع یکی از دو کار زیر را انجام می‌دهد :

- اگر آرگومانی که به تابع ()MessageBeep ارسال می‌شود، مثبت باشد، صدای بوق از طریق کارت صوتی کامپیوتر به گوش می‌رسد.
- اگر آرگومانی که به تابع ( )MessageBeep ارسال می‌شود، منفی باشد، صدای بوق از طریق بلندگوی کامپیوتر به گوش می‌رسد.

یک پروژه ایجاد کنید که در آن تنها یک دکمه در وسط فرم قرار داشته باشد. نام دکمه را با cmdBeep تنظیم کنید و عنوان آن را به Beep تغییر دهید. روی دکمه کلیک کنید تا صدایی از طریق کارت صوتی یا بلندگوی سیستم پخش شود.

۲ــ فرمی به شکل زیر طراحی کرده و مشخصه‌های آن را تنظیم کنید. با کلیک روی دکمه‌های Start Timing و End Timing زمان سیستم خوانده شده و در برچسب‌های مقابل آن‌ها نمایش داده می‌شود. سپس اختلاف آن‌ها محاسبه و در برچسب سوم نمایش داده می‌شود.



شکل ۴ــ۳

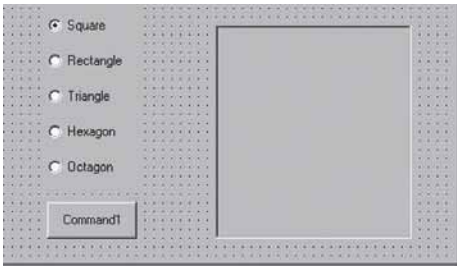
۳ــ ترسیم بیضی: برنامه کاربردی جدیدی را شروع کنید. مشخصه‌ی ScaleMode را با Pixels مقداردهی کنید. به کمک توابع APIکدی بنویسید که با تغییر اندازه‌ی فرم، بیضی رسم شود.

۴ــ ترسیم خط: برنامه کاربردی جدیدی را شروع کنید. یک دکمه‌ی فرمان

در فرم قرار دهید. مشخصه‌ی ScaleMode فرم را Pixels انتخاب کنید. با کلیک روی یک دکمه‌ی فرمان، بیضی رسم شود.

۵ــ ترسیم چند ضلعی‌ها

برنامه‌کاربردی جدیدی را شروع کرده و فرمی با کنترل‌های زیر ایجادکنید : یک کادر تصویر (ScaleMode با Pixels مقداردهی شده است)، آرایه‌ی کنترلی از پنج دکمه‌ی انتخاب و یک دکمه‌ی فرمان.با انتخاب هر کدام از گزینه‌ها و کلیک روی دکمه‌ی فرمان، چند ضلعی مورد نظر در کادر تصویر ترسیم شود.

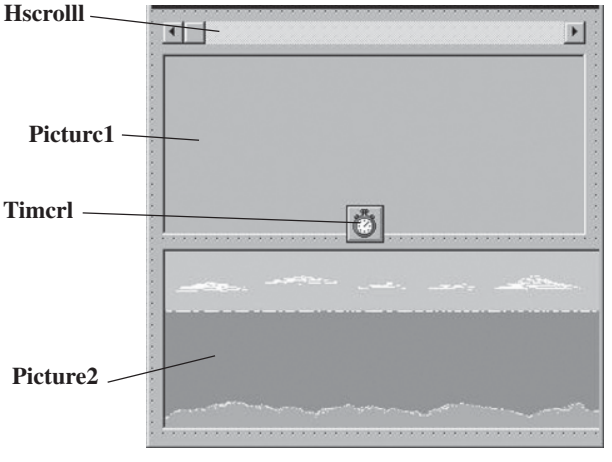


شکل۵ــ۳

۶ــ اسکرول‌کردن افقی زمینه

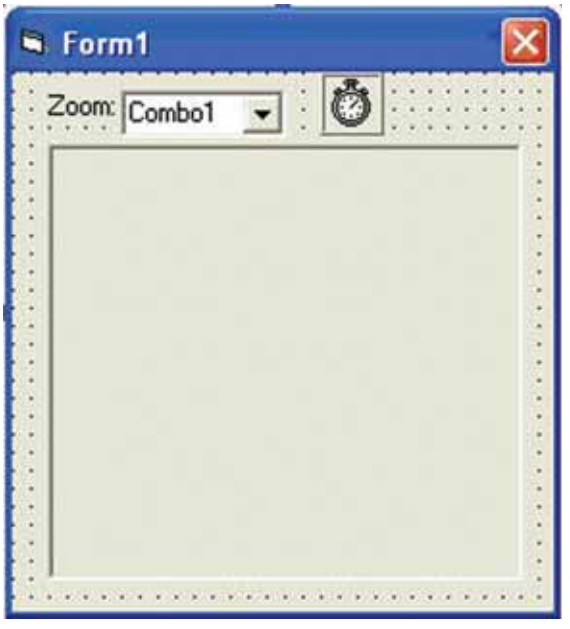
برنامه کاربردی جدیدی را شروع کنید. نوار لغزان افقی، دو کادر تصویر و کنترل تایمر اضافه کنید. فرم شبیه شکل زیر خواهد بود :

کدی بنویسید که با اسکرول کردن، تصویر حرکت کند.



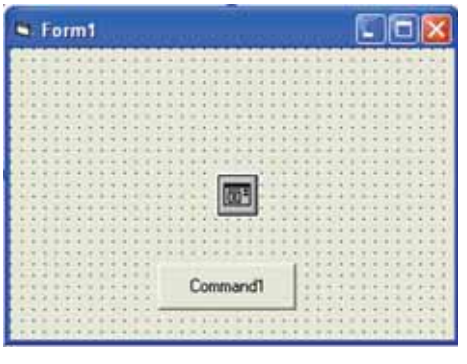
شکل۶ــ۳

۷ــ این تمرین یک Magnifier یا ذره‌بین می‌باشد که تصویر زیر اشاره‌گر ماوس را بزرگ کرده و در شی، Picture کپی می‌کند.



شکل۷ــ۳

۸ ــ به وسیله‌ی شیء CommonDialog یک فایل صوتی از نوع WAVE را انتخاب و به وسیله تابع SndPlay Sound آن را پخش کنید.



شکل۸ــ۳

۹ــ این مثال، تصویر زمینه دسک‌تاپ و همچنین تصویر کل صفحه نمایش را در یک شیء تصویر کپی می‌کند.

شکل ۹ــ۳