

یافتن یک تراک خاص از CD

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
3.1	2, 3	۲۱ جون ۱۹۹۵	112732.TXT

ابزار MCI این قابلیت را دارد که توسط خصوصیت های To و Command یک تراک مشخص از CD را بیابد. برای یافتن یک تراک خاص از CD شماره آن تراک را در خصوصیت To قرار دهید. سپس با کمک خصوصیت Command عمل جستجو را انجام دهید.

خصوصیت To از ابزار MCI این امکان را فراهم می آورد که بر مبنای فرمت جاری زمان، روی CD جابجا شوید. اگر به خصوصیت TimeFormat مقدار MCI_FORMAT_TMSF (یا مقدار ۱۰) بدهید، این امکان فراهم می شود که به ابتدای تراک خاصی منتقل شوید.

مثال زیر نشان می دهد که چگونه توسط ابزار MCI از ابتدای هر یک از تراک های یک CD

مدت ۵ ثانیه را نواخته و به سراغ بعدی بروید.

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

۲- یک ابزار MCI با نام MMControl1 به Form1 اضافه نمایید.

۳- یک CommandButton نیز روی این فرم قرار دهید.

۴- کد زیر را در واقعه Form_Load از Form1 قرار دهید:

```
Sub Form_Load()
    MMControl1.DeviceType = "CDAudio"
    MMControl1.Command = "Open"
End Sub
```

۵- کد زیر را در واقعه Command1_Click وارد نمایید:


```
Sub Command1_Click()
    ' Set the timeformat to allow seek to move between tracks:
    mmcontrol1.TimeFormat = 10
    ' Loop through all tracks and play each for five seconds:
    For i = 1 To MMControl1.Tracks
        MMControl1.To = Str$(i)
        MMControl1.Command = "Seek"
        MMControl1.Command = "Play"
        x = Timer
    Next i
End Sub
```

```

While Timer < x + 5
    DoEvents
Wend
Next
' Stop the CD:
MMControl1.Command = "pause"
End Sub

```

۶- برنامه را اجرا کنید. ابزار MCI باید شروع به نواختن تراک دوم نماید.

توجه:  پیش از اجرای این برنامه، باید یک CD در درایو مربوطه قرار گرفته و آماده نواختن باشد. بنابراین باید کدی به برنامه خود اضافه نمایید که تشخیص دهد که یک CD وارد درایو شده است یا خیر، و پس از آن برنامه را اجرا کند.

یافتن زمان کل موسیقی موجود روی CD

نام فایل	تاریخ	نگارش ویدئو/بیسیک	نگارش ویندوز
112766.TXT	۲۱ جون ۱۹۹۵	3	3.1

رابط برنامه نویسی چند رسانه ای و ابزار MCI دارای فرامینی هستند که می توانند زمان کل موسیقی موجود روی CD را مشخص کنند. خصوصیت Length مربوط به ابزار MCI می تواند طول زمان رسانه جاری (مثلا تراک جاری) را مشخص کند. برای این کار ابتدا خصوصیت TimeFormat از ابزار MCI را معادل مقدار MCI_FORMAT_TMSF قرار داده و با جمع زدن مقادیر حاصل توسط تابع DateAdd می توانید زمان کل را بدست آورید.

تابع DateAdd عمل جمع زدن را از 12:00:00 am آغاز می کند، بنابراین باید کنترل کنید که کل زمان از یک ساعت کمتر است یا خیر. اگر کمتر از یک ساعت بود کافیسست سه کاراکتر

اول (:12) را از ابتدای آن و حروف AM را از انتهای آن حذف کنید. اگر زمان کل از یکساعت بیشتر بود باید حروف AM را از انتهای آن حذف کنید تا به این شکل در آید:

1:09:43

برای ساختن مثال به ترتیب زیر عمل کنید:

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

۲- یک TextBox به نام Text1 و یک ابزار MCI به نام MMControl1 به Form1 اضافه کنید.

۳- کد زیر را در واقعه Form_Load از Form1 قرار دهید:

```
Sub Form_Load ()
    ' Initialize the CD:
    mmcontrol1.TimeFormat = 10
    mmcontrol1.DeviceType = "CDAudio"
    mmcontrol1.Command = "Open"
    ' Dimension variables:
    Dim Length As Variant
    Dim CDSeconds As Integer, CDMinutes As Integer
    ' Calculate minutes and seconds:
    CDSeconds = CDSeconds + (mmcontrol1.Length And &HFF00&) / &H100
    CDMinutes = CDMinutes + (mmcontrol1.Length And &HFF)
    ' Sum minutes and seconds:
    Length = DateAdd("s", CDSeconds, Length) ' Add Seconds
    Length = DateAdd("n", CDMinutes, Length) ' Add Minutes
    ' Determine if total running time is less than one hour:
    If (Left$(Length, 2)) = "12" Then ' Less than 1 hour
        Text1.Text = Mid(Length, 4, (Len(Length) - 4))
    Else ' Greater than 1 hour:
        Text1.Text = Left(Length, (Len(Length) - 3))
    End If
End Sub
```

۴- برنامه را اجرا کنید. زمان کل موسیقی موجود روی دیسک در Text1 نمایش داده می شود.

توجه:  پیش از اجرای این برنامه باید یک CD درون درایو مربوطه قرار دهید. بنابراین باید

کدی به برنامه خود اضافه کنید که وجود CD در درایو را تشخیص داده و سپس برنامه را اجرا کند.

تغییر میزان بلندی صدای ویندوز			
نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
118377.TXT	۲۱ می ۱۹۹۸	3	3.1

ابزار MCI امکان تغییر میزان بلندی صدای ویندوز را فراهم نمی آورد. برای تغییر میزان بلندی صدا باید از API چند رسانه ای و تابع WaveOutSetVolume استفاده نمایید. ضمناً برای تعیین میزان بلندی صدا می توانید تابع WaveOutGetVolume را بکار برید.

تابع WaveOutSetVolume میزان بلندی صدای Wave را تنظیم می کند. تنظیم بلندی صدای کانالهای چپ و راست توسط این تابع امکان پذیر است. برای تعریف این تابع دو روش وجود دارد. در روش اول میزان بلندی صدای هر دو کانال همزمان و در یک پارامتر برای تابع ارسال می شود:

```
'Enter each Declare statement as one, single line:
Declare Function waveOutSetVolume Lib "MMSYSTEM.DLL"
    (ByVal wDeviceID as Integer, ByVal dwVolume as Long) as Integer
Declare Function waveOutGetVolume Lib "MMSYSTEM.DLL"
    (ByVal wDeviceID as Integer, dwVolume as Long) as Integer
```

روش دوم این امکان را فراهم می کند که بتوان هر یک از کانالهای چپ و راست را جداگانه برای تابع ارسال کرد:

```
'Enter the following Declare statement as one, single line:
Declare Function waveOutSetVolume Lib "MMSYSTEM.DLL"
    (ByVal wDeviceID as Integer, ByVal dwVolumeRight as Integer
    ByVal dwVolumeLeft as Integer) as Integer
```

پارامترهای این تابع عبارتند از:

wDeviceID : این پارامتر تعیین کننده شماره دستگاه خروجی از نوع wave است. اگر بخواهید تنها یک فایل wave را بنوازید، باید این مقدار را صفر قرار دهید. برای پخش فایل‌های بعدی به ترتیب ۱ و ۲ و ... را استفاده کنید.

DwVolume : این پارامتر در تابع waveOutSetVolume مشخص کننده مقدار جدید بلندی صدا است. در تابع waveOutGetVolume این پارامتر مشخص کننده اشاره گر به محل ذخیره میزان فعلی بلندی صدا می باشد.

دو بایت پایین این پارامتر چهار بایتی نشان دهنده میزان بلندی صدای کانال چپ بوده و دو بایت بالای آن حاوی میزان بلندی صدای کانال راست می باشد. برای هر یک از این کانال ها مقدار &HFFFF نشان دهنده حداکثر بلندی صدا و مقدار &H0000 نشان دهنده حداقل بلندی صدا است. اگر دستگاهی از هر دو کانال چپ و راست پشتیبانی نکند، فقط دو بایت پایین پارامتر dwVolume مورد استفاده قرار خواهد گرفت.

در صورتیکه تابع با موفقیت عمل خود را انجام دهد، مقدار برگشتی این پارامتر صفر خواهد بود. در غیر اینصورت مقدار برگشتی مشخص کننده کد خطا خواهد بود که یکی از موارد زیر است:

MMSYSERR_INVALIDHANDLE = 5 شناسه دستگاه (Device Handle) نادرست بوده

MMSYSERR_NOTSUPPORTED = 8 این تابع روی این سیستم پشتیبانی نمی شود

MMSYSERR_NODRIVER = 6 درایور مربوطه نصب نگردیده است

تمامی دستگاهها امکان تغییر میزان بلندی صدا را در هر دو کانال چپ و راست ندارند. بیشتر دستگاهها از تغییر تمام ۱۶ بیت میزان بلندی صدا پشتیبانی نمی کنند و بنابراین از بیتهای بالا استفاده نمی کنند.

مثال زیر روش استفاده از توابع waveOutSetVolume و waveOutGetVolume را نشان

می دهد:

- ۱- یک پروژه جدید در ویژوال بیسیک ایجاد کنید. Form1 بطور پیش فرض ایجاد می شود.
- ۲- ابزارهای زیر را به Form1 اضافه کنید و خصوصیتهای آنها را بصورتی که مشخص شده است تنظیم نمایید:

ابزار	خصوصیت	مقدار
Command Button	Name	PlaySound
	Caption	Play Sound
Command	Name	LeftUp

Button	Caption	Left Up
Command	Name	RightUp
Button	Caption	Right Up
Command	Name	RightDown
Button	Caption	Right Down
Label	Caption	LeftVol
Label	Caption	RightVol

۳- کد زیر را به بخش declarations از Form1 اضافه نمایید:

```
' Enter each of the following Declare statements as one, single line:
  Declare Function sndPlaySound Lib "MMSYSTEM.DLL"
    (ByVal lpszSoundName As String, ByVal wFlags As Integer) As Integer
  Declare Function waveoutSetVolume Lib "mmsystem.dll"
    (ByVal wDeviceID As Integer, ByVal dwVolumeRight As Integer,
     ByVal dwVolumeLeft As Integer) As Integer
  Declare Function waveOutGetVolume Lib "MMSYSTEM.DLL"
    (ByVal wDeviceID As Integer, lpdwvolume As Long) As Integer

  Const SND_ASYNC = &H1
  Const SND_NODEFAULT = &H2

  Dim CurrentVolLeft As Long
  Dim CurrentVolRight As Long
```

۴- کد زیر را در Form1 وارد نمایید:

```
Sub Form_Load ()
  Dim x As Integer
  Dim BothVolumes As Long
  ' Note that the waveid is 0 indicating the first wave output device.
  ' If you were to play multiple wavefiles on multiple wave output devices
  ' you would use 1 for the second wave output device, 2 for the third and
  ' so on.
  ' This code will retrieve the current volume setting
  x = waveOutGetVolume(0, BothVolumes)

  ' This code isolates the low-order word.
  ' Note that the value &HFFFF& is a Long Integer, which is the same
  ' as 0000FFFF, but because Visual Basic would automatically
  ' truncate this to FFFF, you must force the logical operation to use
  ' a four-byte Long Integer (0000FFFF) rather than a two-byte Integer
  ' (FFFF). This is accomplished by using the type casting
  ' character (&).
  CurrentVolLeft = BothVolumes And &HFFFF&

  ' This code isolates the high-order word.
  ' Enter the following two lines as one, single line:
  CurrentVolRight = ((BothVolumes And &HFFFF0000) / &H10000)
    And &HFFFF&

  LeftVol.Caption = Hex$(CurrentVolLeft) ' Update the label with the
  RightVol.Caption = Hex$(CurrentVolRight) ' current volume settings.
End Sub
```

```
Sub PlaySound_Click ()
    Dim x As Integer
    Dim wFlags As Integer
    Dim SoundName As String
    SoundName = "C:\WINDOWS\MSREMIND.WAV" ' Pick any wave file.
    wFlags = SND_ASYNC Or SND_NODEFAULT
    x = sndPlaySound(SoundName$, wFlags%) ' Play the wave file.
End Sub

Sub LeftUp_Click ()
    ' Increase the left sound channel setting:
    Dim x As Integer
    CurrentVolLeft = CurrentVolLeft + &H1000&
    ' Prevent the channel setting from exceeding the maximum limit:
    If CurrentVolLeft > &HFFFF& Then CurrentVolLeft = &HFFFF&
    LeftVol.Caption = Format$(Hex$(CurrentVolLeft))
    ' Enter the following two lines as one, single line:
    x = waveoutSetVolume(0, Cint("&H" + Hex$(CurrentVolRight)),
        Cint("&H" + Hex$(CurrentVolLeft)))
End Sub

Sub LeftDown_Click ()
    ' Decrease the left sound channel setting:
    Dim x As Integer
    CurrentVolLeft = CurrentVolLeft - &H1000&
    ' Prevent the channel setting from dropping below the minimum limit:
    If CurrentVolLeft < &H0& Then CurrentVolLeft = &H0&
    LeftVol.Caption = Hex$(CurrentVolLeft)
    ' Enter the following two lines as one, single line:
    x = waveoutSetVolume(0, Cint("&H" + Hex$(CurrentVolRight)),
        Cint("&H" + Hex$(CurrentVolLeft)))
End Sub

Sub RightUp_Click ()
    ' Increase the right sound channel setting:
    Dim x As Integer
    CurrentVolRight = CurrentVolRight + &H1000&
    ' Prevent the channel setting from exceeding the maximum limit.
    If CurrentVolRight > &HFFFF& Then CurrentVolRight = &HFFFF&
    RightVol.Caption = Hex$(CurrentVolRight)
    ' Enter the following two lines as one, single line:
    x = waveoutSetVolume(0, Cint("&H" + Hex$(CurrentVolRight)),
        Cint("&H" + Hex$(CurrentVolLeft)))
End Sub

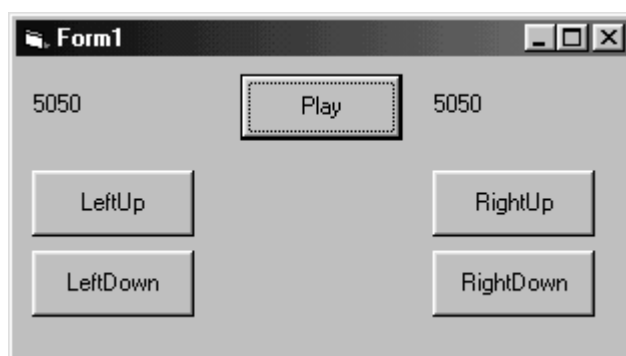
Sub RightDown_Click ()
    ' Decrease the right sound channel setting:
    Dim x As Integer
    CurrentVolRight = CurrentVolRight - &H1000&
    ' Prevent the channel setting from dropping below the minimum limit:
    If CurrentVolRight < 0 Then CurrentVolRight = 0
    RightVol.Caption = Hex$(CurrentVolRight)
    ' Enter the following two lines as one, single line:
    x = waveoutSetVolume(0, Cint("&H" + Hex$(CurrentVolRight)),
        Cint("&H" + Hex$(CurrentVolLeft)))
End Sub
```


۵- برنامه را اجرا نمایید.

۶- مقدار فعلی بلندی صدا را در Label ها خواهید دید. روی دکمه Play Sound کلیک کنید تا یک فایل wave با میزان فعلی بلندی صدا نواخته شود.

۷- روی دیگر دکمه ها کلیک کنید تا میزان صدای کانالهای چپ یا راست تغییر کند و سپس روی Play Sound کلیک نمایید. میزان بلندی صدا تغییر کرده است.

برای کسب اطلاعات بیشتر در مورد این توابع و دیگر توابع مربوط به چندرسانه ای مراجعه کنید به "Multimedia Programmer's Reference Guide" که بخشی از SDK مخصوص ویندوز ۳/۱ است. نحوه تعریف کردن توابع دیگر چندرسانه ای و مقادیر ثابت مورد نیاز هر یک را می توانید در فایل WINMMSYS.TXT که بخشی از نگارش حرفه ای ویژوال بیسیک ۳ است بیابید.



شکل 1 - نمایی از برنامه تغییر شدت صدا

تذکر مترجم : از آنجا که مثال فوق به راحتی در نگارش های جدید ویژوال بیسیک قابل اجرا نیست تغییراتی در آن دادم تا بتوان آنرا به راحتی اجرا نمود. از توضیحات اضافی خودداری کرده و فقط مراحلی که در آنها تغییری بوجود آمده است را ارایه می دهم:
 کد مورد استفاده در مرحله ۳ به این صورت تغییر کند :

```
Private Declare Function sndPlaySound Lib "winmm.dll" Alias _
    "sndPlaySoundA" (ByVal lpszSoundName As String, _
    ByVal uFlags As Long) As Long
Private Declare Function waveOutSetVolume Lib "winmm.dll" _
```

```
(ByVal uDeviceID As Long, ByVal dwVolume As Long) As Long
Private Declare Function waveOutGetVolume Lib "winmm.dll" _
    (ByVal uDeviceID As Long, lpdwVolume As Long) As Long

Const SND_ASYNC = &H1
Const SND_NODEFAULT = &H2

Dim CurrentVolLeft As Long
Dim CurrentVolRight As Long
```

به علاوه باید فراخوانی تابع waveOutSetVolume بصورت زیر انجام شود:

```
x = waveOutSetVolume(0, CLng("&H" + Hex$(CurrentVolRight) + _
    Hex$(CurrentVolLeft)))
```

نمایش متن روی تصاویر AVI در حال نمایش

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
3.1	3	۱۴ آگوست ۱۹۹۵	131819.TXT

این بخش نشان می دهد که چگونه متنی را روی یک AVI در حال پخش نمایش دهیم. برای اجرای این مثال باید درایورهای Video for Windows روی کامپیوتر شما نصب شده و یک فایل AVI مناسب در اختیار داشته باشید.

زمانی که برای نمایش یک AVI از Picture Box استفاده می کنید، ابزار MCI پس از نمایش هر فریم از فایل AVI محتویات Picture Box را پاک می کند. این باعث می شود که هر متنی که روی Picture Box نمایش داده شده است از بین برود. مثال زیر روشی را نشان می دهد که در آن به کمک یک Timer متن مورد نظر در اسرع وقت بازنویسی شود. این روش باعث می شود که متن در هنگام پاک شدن محتویات Picture Box برای نمایش فریم بعد، دوباره نوشته شده و فقط کمی حالت چشمک زن پیدا کند.

۱- پروژه زیر را در محیط ویژوال بیسیک بازیابی کنید:

VB\Samples\MCI\Mcitest.mak.

۲- تغییرات گفته شده در بخش زیر را در آن اعمال کنید تا بتواند فایل AVI را نمایش دهد:

```
ARTICLE-ID:Q98769
TITLE      :Playing an .AVI File with the MCITEST Example
```

۳- در پنجره Animate.frm یک Timer اضافه کنید.

۴- کد زیر را در واقعه Timer1_Timer وارد کنید:

```
Sub Timer1_Timer ()
    Dim MyStr As String
    MyStr = "Testing Testing" 'String you want to display
    picture1.CurrentX = 10
    picture1.CurrentY = 10
    picture1.Print MyStr$
End Sub
```

۵- کد موجود در واقعه AI_OPEN را بصورت زیر تغییر دهید:

قبل از تغییر:

```
' Play the movie into the picture control.
On Error GoTo MCI_ERROR
mmcontrol1.Command = "Play"
```

بعد از تغییر:

```
' Play the movie into the picture control.
On Error GoTo MCI_ERROR
Timer1.Interval = 50 'This number is the refresh rate
mmcontrol1.Command = "Play"
```

۶- برنامه را اجرا کنید.

در روش دیگر، از تابع TextOut برای نمایش متن روی اشیایی که دارای دستور Print

نیستند اما خصوصیت hDCs دارند، استفاده می شود:

۱- پروژه زیر را در محیط ویژوال بیسیک بازیابی کنید:

VB\Samples\MCI\Mcitest.mak.

۲- تغییرات گفته شده در بخش زیر را در آن اعمال کنید تا بتواند فایل AVI را نمایش دهد:

```
ARTICLE-ID:Q98769
TITLE      :Playing an .AVI File with the MCITEST Example
```

۳- در پنجره Animate.frm یک Timer اضافه کنید.

۴- کد زیر را در واقعه Timer1_Timer وارد کنید:

```
Sub Timer1_Timer ()
    DimRetVal As Integer 'The return value of TextOut - Ignored.
    Dim xPos As Integer, yPos As Integer
    Dim MyStr As String, MyLen As Integer

    xPos = 10 'x coordinate where you want the text to display
    yPos = 10 'y coordinate where you want the text to display
    MyStr = "Testing Testing" 'String you want to display
    MyLen = Len(MyStr) 'Length of string

    'Display the text on top of picture1 (location where AVI is playing)
   RetVal= TextOut(Picture1.hDC, xPos, yPos, MyStr, MyLen)
End Sub
```

۵- کد موجود در واقعه AI_OPEN را بصورت زیر تغییر دهید:

قبل از تغییر:

```
' Play the movie into the picture control.
On Error GoTo MCI_ERROR
mmcontrol1.Command = "Play"
```

بعد از تغییر:

```
' Play the movie into the picture control.
On Error GoTo MCI_ERROR
Timer1.Interval = 50 'This number is the refresh rate
mmcontrol1.Command = "Play"
```

۶- کد زیر را به بخش Declarations از ماژول Global.BAS اضافه کنید:

```
' Enter the following Declare statement as one, single line:
Declare Function TextOut Lib "GDI" (ByVal hDC As Integer, ByVal X As
    Integer, ByVal Y As Integer, ByVal lpString As String, ByVal nCount
    As Integer) As Integer
```

۷- برنامه را اجرا کنید.

شبیه سازی دستور SOUND کوئیک بیسیک در ویژوال بیسیک

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
140879.TXT	۲۱ می ۱۹۹۸	4 (16 bit)	3.1

دستور SOUND که در کوئیک بیسیک به کار می رفت در ویژوال بیسیک معادلی ندارد. برای اینکه عملکرد این دستور را در محیط ویندوز شبیه سازی کنید باید از یک تابع API استفاده نمایید.

برای اینکار لازمست توسط فراخوانی روتین OpenSound صفی از اصوات ایجاد کنید. با کمک تابع SetVoiceSound تمام مقادیر مورد نیاز فرکانس و مدت زمان را در صف فوق قرار دهید. هنگامی که صف صوت مذکور از اطلاعات لازم پر شد، با فراخوانی StartSound پردازش صف را آغاز کنید. پس از اینکه اصوات نواخته شدند، توسط تابع CloseSound صف را از بین ببرید.

اگر قصد دارید اطلاعات خیلی زیادی در صف قرار دهید ممکنست لازم شود توسط SetVoiceQueueSize طول صف (بافر) را افزایش دهید.

پس از اجرای تابع StartSound نمی توانید اصوات بیشتری در همان صف قرار دهید مگر اینکه صف را خالی کنید. قرار دادن اصوات جدید در صف باعث می شود این اطلاعات جدید روی اطلاعات قبلی نوشته شوند. در صورتیکه قصد دارید پس از اجرای دستور StartSound اصوات جدیدی در صف قرار دهید، لازمست تابع WaitSoundState را به همراه پارامتر ۱ اجرا نمایید. هنگامی که تابع مذکور مقدار Null برگرداند، نشان دهنده اینست که صف اصوات خالی است و پروسه برنامه می تواند ادامه یابد.

مثال زیر که شبیه سازی برای دستور Sound کوئیک بیسیک است، نحوه استفاده از توابع API را نشان می دهد.

در بخش Declarations برنامه باید کد زیر قرار بگیرد:

```
Public Declare Function OpenSound Lib "sound.drv" () As Integer
Public Declare Function VoiceQueueSize Lib "sound.drv"
    (ByVal nVoice%, ByVal nBytes%) As Integer
```

```
Public Declare Function SetVoiceSound Lib "sound.drv"
    (ByVal nSource%, ByVal Freq%, ByVal nDuration%) As Integer
Public Declare Function StartSound Lib "sound.drv" () As Integer
Public Declare Function CloseSound Lib "sound.drv" () As Integer
Public Declare Function WaitSoundState Lib "sound.drv" (ByVal State%)
As Integer
```

توجه: تمامی دستورات Declare فوق باید در یک سطر قرار بگیرند. 

تابع SetVoiceSound دو پارامتر ورودی دارد. پارامتر اول یعنی Freq یک پارامتر چهار بایتی است. دو بایت بالا میزان فرکانس بر مبنای هرتز است و دو بایت پایین کسر فرکانس را نشان می دهد. فرمول $X * 2^{12}$ مقدار X را به دو بایت بالا شیفیت می دهد. پارامتر دوم یا Duration% مدت زمان نواخته شدن صدا بر حسب واحد "تیک ساعت" است. در پردازنده های اینتل هر ثانیه معادل $18/2$ تیک ساعت است.

مثال ساده زیر نشان می دهد که چگونه تعدادی فرکانس و مدت در صف قرار داده و توسط

فراخوانی تابع StartSound این اصوات را ایجاد کنید:

```
Private Sub Form_Click ()
    Suc% = OpenSound()
    S% = SetVoiceSound(1, 100 * 2 ^ 16, 100) ' Frequency = 100 hz
    S% = SetVoiceSound(1, 90 * 2 ^ 16, 90) ' Frequency = 90 hz
    S% = SetVoiceSound(1, 80 * 2 ^ 16, 90) ' Frequency = 80 hz
    S% = StartSound()
    While (WaitSoundState(1) <> 0): Wend ' Wait for sound to play.
    Succ% = CloseSound()
End Sub
```

مثال دیگری که در زیر ملاحظه می کنید، صدای آژیر پلیس را ایجاد می کند:

۱- این روتین را در بخش General وارد کنید:

```
Public Sub Sound (ByVal Freq as Long, ByVal Duration%)
    Freq = Freq * 2 ^ 16 ' Shift frequency to high byte.
    S% = SetVoiceSound(1, Freq, Duration%)
    S% = StartSound()
    While (WaitSoundState(1) <> 0): Wend
End Sub
```

۲- کد زیر را در برنامه وارد کنید. در این مثال واقعه Form_Click بکار رفته تا کلیک

کردن روی هر نقطه از فرم باعث ایجاد صدای آژیر شود:

```
Private Sub Form_Click ()
    Suc% = OpenSound()
    For j& = 440 To 1000 Step 5
        Call Sound(j&, j& / 100)
    Next j&
    For j& = 1000 To 440 Step -5
```

```

Call Sound(j&, j& / 100)
Next j&
Succ% = CloseSound()
End Sub

```

نمایش فایل AVI در حالت تمام صفحه

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
142389.TXT	۲۱ می ۱۹۹۸	4	3.1

در این بخش به کمک یک مثال خواهیم دید که چگونه یک فایل ویدیویی AVI را در حالت تمام صفحه نمایش دهیم. هنگامی که یک فایل AVI را به صورت تمام صفحه نمایش دهید، تخته رنگ^۱ جاری سیستم فقط از تصاویر فایل AVI تبعیت می کند. از آنجا که هیچ پنجره دیگری روی صفحه وجود ندارد، عمل سازگاری رنگها^۲ نیز صورت نمی گیرد.

مثال زیر از تابع `mciSendString` ویندوز استفاده می کند. برای اینکه بتوانید مثال را اجرا کنید لازمست کامپیوتر شما قابلیت نمایش فایل‌های AVI را داشته باشد.

فایل AVI مثال زیر یکی از فایل‌های همراه `Video for Windows` است. می توانید نام و مسیر این فایل را با هر فایل AVI دیگری که در اختیار دارید عوض کنید.

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

۲- یک `CommandButton` به نام `Command1` ایجاد کنید.

۳- سطرهای زیر را به بخش `Declarations` اضافه کنید:

```

#If Win32 Then
Private Declare Function mciSendString Lib "winmm.dll" Alias _
    "mciSendStringA" (ByVal lpstrCommand As String, ByVal _
    lpstrReturnString As Any, ByVal uReturnLength As Long, ByVal _
    hwndCallback As Long) As Long

```

^۱ Color Palette

^۲ Dithering of Colors

```
#Else
Private Declare Function mciSendString Lib "mmsystem" (ByVal _
lpstrCommand As String, ByVal lpstrReturnStr As Any, ByVal _
wReturnLen As Integer, ByVal hCallBack As Integer) As Long
#End If
```

۴- کد زیر را به واقعه Command1_Click اضافه کنید:

```
Sub Command1_Click ()
CmdStr$ = "play c:\winvideo\wndsurf1.avi fullscreen "
ReturnVal& = mciSendString(CmdStr$, 0&, 0, 0&)
End Sub
```

۵- برنامه را اجرا کنید.

روی دکمه Command1 کلیک کنید تا ویدیوی مذکور بصورت تمام صفحه پخش شود.

تعیین برچسب منحصر بفرد برای CD			
نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
147654.TXT	۷ آگوست ۱۹۹۸	4 , 5 , 6	95 / 98

بخشی از استاندارد چندرسانه ای در مورد CD های صوتی تعیین کرده است که همگی این CD ها باید دارای یک برچسب منحصر بفرد با فرمت زیر باشند:

XXXX-#####

که در آن XXXX کد تولید کننده و ##### یک عدد منحصر بفرد پنج رقمی برای CD است. این اطلاعات در تراک داخلی CD ضبط شده و ضمناً روی لبه حلقه داخلی CD نیز حک می شود. نه تمامی تولید کنندگان دارای کد مخصوص هستند و نه همگی از این فرمت پیروی می کنند. اگرچه این روش یک استاندارد است، API تابع خاصی برای گرفتن این اطلاعات ندارد. در این بخش روش خواندن یا ایجاد کردن چنین اطلاعاتی را از طریق برنامه نویسی خواهیم دید. برای خواندن مشخصه منحصر بفرد از CD باید MSCDEX را مستقیماً فراخوانید. اطلاعات بیشتری در زمینه فراخوانی MSCDEX را می توانید در مشخصات فنی MSCDEX 2.20 پیدا کنید.

از آنجا که بعضی تولید کنندگان CD این مشخصه منحصر بفرد را روی CD های خود ندارند، مایکروسافت توصیه می کند برنامه نویسان بر مبنای اطلاعات ذخیره شده روی CD چنین شماره منحصر بفردی برای CD در نظر بگیرند.

در مثال زیر روشی برای ایجاد یک عدد منحصر بفرد برای شماره گذاری روی CD های صوتی ارائه می شود. هدف استفاده از این عدد فراهم کردن این امکان برای برنامه های ویژوال بیسیک است که یک CD خاص را تشخیص داده و اطلاعات را از آن بخوانند.

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

۲- یک TextBox به نام Text1 و یک ابزار چندرسانه ای MMControl1 به فرم اضافه کنید.


۳- کد زیر را در واقعه Form_Load قرار دهید:

```
Private Sub Form_Load ()
    Dim DiskID As Long
    Dim Track As Integer
    ' Initialize CD:
    mmcontrol1.DeviceType = "CDAudio"
    mmcontrol1.Command = "open"
    ' Make unique number based on tracks and tracklength:
    DiskID = mmcontrol1.Tracks
    For Track = 1 To mmcontrol1.Tracks
        mmcontrol1.Track = Track
        DiskID = DiskID + mmcontrol1.TrackLength ' Add 4-byte TrackLength
        DiskID = DiskID + mmcontrol1.Length ' Add 4-byte CD Length
    Next Track
    ' Set text to unique value:
    Text1.Text = DiskID
End Sub
```

۴- یک CD درون درایو قرار دهید.

۵- برنامه را اجرا کنید.

۶- Text1 باید محتوی یک کد شناسایی منحصر بفرد برای CD باشد.

توجه:  پیش از اجرای برنامه باید یک CD درون درایو مربوطه قرار داشته باشد. بنابراین باید کدی به برنامه خود اضافه کنید که تشخیص دهد CD درون درایو قرار دارد یا خیر، و سپس برنامه را اجرا کند.

تعیین اینکه آیا کامپیوتر دارای کارت صوتی هست یا خیر

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
3.1 / 95 / 98	4 , 5 , 6	۷ آگوست ۱۹۹۸	147811.TXT

این مثال نشان می دهد که چگونه تعیین کنیم که کامپیوتر دارای کارت صوتی با توان پخش فایل های Wave هست یا خیر.

API تابعی دارد به نام WaveOutGetNumDevs که تعداد دستگاه های موجود در سیستم با توانایی پخش فایل های صوتی Wave را مشخص می کند.

۱- یک پروژه جدید در ویژوال بیسیک ایجاد کنید و یک Command Button روی Form1 قرار دهید.

۲- کد زیر را در بخش Declarations از Form1 قرار دهید.

```
#If Win32 Then
    Private Declare Function waveOutGetNumDevs Lib "winmm" () As Long
#ElseIf Win16 Then
    Private Declare Function waveOutGetNumDevs Lib "mmsystem" () _
        As Integer
#End If
```

۳- کد زیر را به واقعه Command1_Click اضافه کنید:

```
Private Sub Command1_Click()
    #If Win32 Then
        Dim i As Long
    #ElseIf Win16 Then
        Dim i As Integer
    #End If
    i = waveOutGetNumDevs()
    If i > 0 Then ' There is at least one device.
        MsgBox "You Can Play Wave Data"
    Else
        MsgBox "Cannot Play Wave Data"
    End If
End Sub
```

۴- برنامه را اجرا کنید و سپس روی Command Button کلیک کنید. یک پیام روی صفحه ظاهر می شود که مشخص می کند کامپیوتر شما توان پخش فایل های Wave را دارد یا خیر.

پخش فایل‌های MIDI به کمک توابع API

نگارش ویندوز	نگارش ویژوال بیسیک	تاریخ	نام فایل
95 / 98	5 , 6	۷ آگوست ۱۹۹۸	171980.TXT

در این بخش روش پخش فایل‌های MIDI از درون ویژوال بیسیک به کمک تابع `mciSendString` را خواهیم دید. البته برای پخش فایل‌های MIDI می‌توانید از ابزار MCI استفاده کنید، و در آن صورت نیازی به استفاده از توابع API وجود ندارد.

۱- یک پروژه جدید در ویژوال بیسیک ایجاد کنید. Form1 ایجاد خواهد شد.

۲- یک `CommandButton` به فرم اضافه کنید.

۳- کد زیر را به واقعه `Command1_Click` اضافه کنید:

```
Private Sub Command1_Click()
    Dim ret As Integer

    ' The following will open the sequencer with the CANYON.MID
    ' file. Canyon is the device_id.
    ret = mciSendString( _
        "open " & Song & " type sequencer alias canyon", 0&, 0, 0)

    ' The wait tells the MCI command to complete before returning
    ' control to the application.
    ret = mciSendString("play canyon wait", 0&, 0, 0)

    ' Close CANYON.MID file and sequencer device
    ret = mciSendString("close canyon", 0&, 0, 0)
End Sub
```

۴- کد زیر را به بخش `Declarations` از Form1 اضافه کنید:

```
Private Declare Function mciSendString Lib "winmm.dll" Alias _
    "mciSendStringA" (ByVal lpstrCommand As String, ByVal _
    lpstrReturnString As Any, ByVal uReturnLength As Long, ByVal _
    hwndCallback As Long) As Long

' Modify the value of the constant "Song" with your path to "canyon.mid".
Private Const Song As String = "C:\Windows\Media\Canyon.MID"
```

۵- برنامه را اجرا کنید. روی دکمه `Command1` کلیک کنید. توجه کنید که باید تا پایان

نواخته شدن موسیقی صبر کنید!