

Playing WAV Files

A program's user interface can be improved by using short sounds to signal certain events such as a download completing or a command being selected. With the right Windows Multimedia function, playing WAV files from a Visual Basic program is child's play. Of course, the system must be equipped with a sound card for this to work.

The API function you need is called, appropriately enough, PlaySound. Its declaration is shown here:

```
Public Declare Function PlaySound Lib "winmm.dll" Alias "PlaySoundA" (ByVal lpszName As String, ByVal hModule As Long, ByVal dwFlags As Long) As Long
```

'The first argument is the name, including path, of the Wave file to play. The second argument is not used when playing sound files and you should pass a value of zero (this function can also play sounds that are associated with system events, but that is not covered here). The final argument consists of flags that control various aspects of how the function works. For the present purposes, two flags are used. They are (with the constants that are typically used for them:

'SND_ASYNC (value= 1) - play asynchronously, meaning that the function returns while the sound is still playing.

'SND_FILENAME (value = &H20000) - the first argument is a file name.

'Thus, the following code plays the sound in DingDong.wav:

```
'PlaySound "dingdong.wav", CLng(0),SND_ASYNC Or SND_FILENAME
```

'When playing sound files in a program there are two other considerations. First, you want to make sure that the specified Wave file exists or else an error will occur. This can be done with the simple function shown here:

```
Public Function FileExists(FullFileName) As Boolean
```

```
' Passed a filename (with path) returns  
' True if the file exists, False if not.
```

```
Dim s
```

```
s = dir(FullFileName)  
If s = "" Then  
    FileExists = False  
Else  
    FileExists = True  
End If
```

```
End Function
```

Second, the program should have an option setting that lets users turn sounds off. This can be maintained as a global Boolean variable with a name such as gProgramSoundsEnabled. My approach is to deal with both

of these considerations in a function as follows. This code assumes that the Wave files are stored in the application folder.

```
Private Sub PlaySoundX(filename As String)

    ' If sound is enabled and filename exists,
    ' play the specified sound.

    filename = App.path & "\" & filename

    If FileExists(filename) And gProgramSoundsEnabled Then
        PlaySound(filename, CLng(0), SND_ASYNC Or SND_FILENAME)
    End If

End Sub
```

With this function in place, the program can play any Wave file like this:

```
PlaySoundX "DingDong.wav"
```

Another way of playing sounds from a Visual Basic program involves including a multimedia control on a form. When you do not need the extra features of the control, however, the technique presented here is a lot easier.