



Help for Wave

[Properties](#) [Events](#)

Registration Information

Order Form

Getting Custom Controls Written

Description

This control makes it easy for you to play and get information about WAV files. Play WAV files in the background as your program performs other tasks; add recorded speech to your applications; put beeps, whistles, and laser sounds into your games. Wave VBX makes all of this very straight-forward. Set the Filename property and tell it to go. That's it

File Name

WAVE1.VBX

Object Type

MabryWave

Compatibility

VB 2.0 and above

Distribution Note When you develop and distribute an application that uses Wave, you should install the file WAVE1.VBX into the users Windows SYSTEM directory. Wave has version information built into it. So, during installation, you should ensure that you are not overwriting a newer version of Wave.

Properties

All of the properties that apply to this control are in this table. Properties that have special meaning for this control or that only apply to this control are marked with an asterisk (*). Action is the default value for the control.

*Action

Align

Enabled

*Exclusive

*FileLength

*Filename

Index

Left

*Loop

*LoopCount

Name

Parent

*PlayEnd

*PlayStart

*Position

*Status

Tag

Top

Events

All of the events that apply to this control are in this table. Events that have special meaning for this control or that only apply to this control are marked with an asterisk (*).

*PlayDone

*PlayLoop

Action Property

[See Also](#) [Example](#)

Description

Specifies what to do with the current WAV file. Not available at design time.

Usage

[*form.*][*control.*]**Action**[= *setting*]

Remarks

The Action property settings are:

Setting	Action taken
1	<p>Play file. Plays the file currently selected by the <u>Filename</u> property. If the <u>Exclusive</u> property is set to False, then nothing else happens while the file plays.</p> <p>If the <u>Exclusive</u> property is False and the <u>Loop</u> property is True, the file plays over and over until another action is taken. The <u>PlayLoop</u> event will fire everytime the file plays.</p> <p>The <u>PlayStart</u> and <u>PlayEnd</u> properties determine what part of the file to play.</p> <p>When the file finally stops playing, for whatever reason, the <u>PlayDone</u> event fires.</p>
2	<p>Pause playing. This action pauses the playback. This only has an effect if the file is currently playing (in other words, if <u>Status</u> is equal to 1).</p>
3	<p>Resume playing. This action restarts the playback. This only has an effect if the file is currently paused (in other words, if <u>Status</u> is equal to 2).</p>
4	<p>Stop playing. This action stops the current playback. This has no effect if the file is already stopped.</p>

Data Type

Integer (enumerated)

See Also

Properties:

[Exclusive](#)

[Filename](#)

[Loop](#)

[PlayEnd](#)

[PlayStart](#)

[Status](#)

Events:

[PlayDone](#)

[PlayLoop](#)

Exclusive Property

[See Also](#) [Example](#)

Description

Determines whether the WAV file plays in the background.

Usage

[*form.*][*control.*]**Exclusive**[= *boolean*]

Remarks

If this property is set to True, the WAV file plays exclusively. This means that nothing else happens while the WAV file is playing. You effectively give control to Wave until the file is done playing (when the [PlayDone](#) event fires).

If this property is set to False, the WAV file plays in the background. When playing in the background, you can set the [Loop](#) property to True. This will make the WAV file play over and over until told to stop.

Data Type

Integer (boolean)

See Also

Properties:

[Action](#)

[Filename](#)

[Loop](#)

Events:

[PlayDone](#)

FileLength Property

[See Also](#) [Example](#)

Description

Tells the length of the current WAV file.

Usage

[*form.*][*control.*]**FileLength**

Remarks

When there is a WAV file selected (see Filename propFilename), this property holds the length of that file (in milliseconds).

Data Type

Integer (long)

See Also

Properties:

[Filename](#)

Filename Property

[See Also](#) [Example](#)

Description

Determines what file is played.

Usage

[*form.*][*control.*]**Filename**[= *string*]

Remarks

This property determines what file is used. If the file could not be found, a file not found error (7) is generated. After a valid file is loaded, the [FileLength](#) property is set to the length of the file, in milliseconds.

Data Type

String

See Also

Properties:

[Action](#)

[Exclusive](#)

[FileLength](#)

[Loop](#)

Events:

[PlayDone](#)

[PlayLoop](#)

Loop Property

[See Also](#) [Example](#)

Description

Determines whether the WAV file automatically repeats.

Usage

[*form.*][*control.*]**Loop**[= *boolean*]

Remarks

If this property is set to True, the WAV file plays repeatedly, until stopped by the program. This only works when the [Exclusive](#) property is set to False.

Each time the file repeats, the [LoopCount](#) property is increased by one and the [PlayLoop](#) event is fired.

Data Type

Integer (boolean)

See Also

Properties:

[Action](#)

[Exclusive](#)

[Filename](#)

[LoopCount](#)

Events:

[PlayLoop](#)

LoopCount Property

[See Also](#) [Example](#)

Description

Tells how many times a file has played (looping).

Usage

[*form.*][*control.*]**LoopCount**[= *integer*]

Remarks

This property holds the number of times a file has repeated playing (while looping, when the Loop property is True and the Exclusive property is False). This property is initially set to zero (0) when the file first starts playing.

Each time the file loops, this property is increased by one. Then, the PlayLoop event is fired.

Data Type

Integer

See Also

Properties:

[Action](#)

[Exclusive](#)

[Filename](#)

[Loop](#)

Events:

[PlayLoop](#)

PlayEnd Property

[See Also](#) [Example](#)

Description

Determines where in the WAV file to stop playing.

Usage

[*form.*][*control.*]**PlayEnd**[= *long*]

Remarks

This property determines where to stop playing the WAV file when the Play [Action](#) is executed. This property must range from [PlayStart](#) to [FileLength](#). This property is in milliseconds.

If this property is set to -1, the WAV file is played up to its end.

Data Type

Integer (long)

See Also

Properties:

Action

FileLength

Filename

PlayStart

PlayStart Property

[See Also](#) [Example](#)

Description

Determines where in the WAV file to start playing.

Usage

[*form.*][*control.*]**PlayStart**[= *long*]

Remarks

This property determines where to start playing the WAV file when the Play [Action](#) is executed. This property must range from 0 to [FileLength](#). This property is in milliseconds.

If this property is set to -1, the WAV file starts playing at the beginning.

Data Type

Integer (long)

See Also

Properties:

[Action](#)

[FileLength](#)

[Filename](#)

[PlayEnd](#)

Position Property

[See Also](#) [Example](#)

Description

Determines the current position in the WAV file.

Usage

[*form.*][*control.*]**Position**[= *long*]

Remarks

This property determines the position of the playback pointer in the WAV file ([Filename](#)). This property is in milliseconds.

You can set this property. But, if the file is playing when you set it, the playing stops.

Data Type

Integer (long)

See Also

Properties:

Action

Filename

Status Property

[See Also](#) [Example](#)

Description

Holds the current status of the control. Not available at design time.

Usage

[*form.*][*control.*]**Status**

Remarks

This property can be one of the following values:

Value	Meaning
1	Currently playing a file (see Filename)
2	Paused
4	Stopped

Data Type

Integer (enumerated)

See Also

Properties:

Action

Filename



Action Example

In this example, the program shows how to play a wave file. It plays the file shown in the edit box. You can pause and stop the file while playing using the buttons. To try this example, paste the code into the Declarations section of a form that contains two labels, two text boxes, three command buttons, and a Wave control. Press F5 and play with the buttons.

```
Option Explicit
```

```
Dim fChanged As Integer ' flag: has filename changed?
```

```
Sub Command1_Click ()  
    ' make sure the filename is up to date  
    SetFilename  
  
    ' play the file  
    Wave1.Action = 1  
    EnableButtons  
End Sub
```

```
Sub Command2_Click ()  
    ' if the user wants to pause  
    If Command2.Caption = "Pause" Then  
        ' pause  
        Wave1.Action = 2  
    Else  
        ' resume  
        Wave1.Action = 3  
    End If
```

```
    EnableButtons  
End Sub
```

```
Sub Command3_Click ()  
    ' stop playing  
    Wave1.Action = 4  
  
    ' update buttons  
    EnableButtons  
End Sub
```

```
Sub EnableButtons ()  
    Select Case Wave1.Status  
        Case 1: ' playing  
            If Command2.Caption <> "Pause" Then  
                Command2.Caption = "Pause"  
            End If  
  
            Command3.Enabled = True  
            Command2.Enabled = True  
  
        Case 2: ' paused  
            If Command2.Caption <> "Resume" Then  
                Command2.Caption = "Resume"
```



```

        End If

        Command3.Enabled = True
        Command2.Enabled = True

    Case 4: ' stopped
        If Command2.Caption <> "Pause" Then
            Command2.Caption = "Pause"
        End If

        Command3.Enabled = False
        Command2.Enabled = False
    End Select
End Sub

Sub SetFilename ()
    ' if the filename has changed, tell Wave VBX
    If fChanged Then
        ' changed no more
        fChanged = False
        Wave1.FileName = Text1

        ' set the file length field
        Text2 = Format(Wave1.FileLength / 1000, "0.000") & " seconds"
    End If
End Sub

Sub Text1_Change ()
    fChanged = True
End Sub

Sub Wave1_PlayDone ()
    EnableButtons
End Sub

Sub Form_Load ()
    ' set-up form and all of its controls
    Form1.BackColor = &H00C0C0C0&
    Form1.BorderStyle = 3 'Fixed Double
    Form1.Caption = "Wave VBX Action Sample"
    Form1.Height = 2955
    Form1.Width = 4215

    Text1.Height = 285
    Text1.Text = "C:\WINDOWS\RINGOUT.WAV"
    Text1.Width = 3615

    Text2.Height = 285
    Text2.Text = ""
    Text2.Width = 3615

    Command1.Caption = "Play"
    Command1.Height = 615
    Command1.Width = 1215

    Command2.Caption = "Pause"
    Command2.Height = 615

```

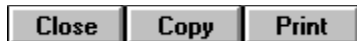
```
Command2.Width = 1215

Command3.Caption = "Stop"
Command3.Height = 615
Command3.Width = 1215

Label1.BackColor = &H00C0C0C0
Label1.Caption = "Filename:"
Label1.Height = 255
Label1.Width = 1935

Label2.BackColor = &H00C0C0C0
Label2.Caption = "File Length:"
Label2.Height = 255
Label2.Width = 2415

fChanged = True
EnableButtons
End Sub
```



Exclusive and Loop Properties Example

In this example, the program shows the effects of the Exclusive and Loop properties. It plays the file shown in the edit box. To try this example, paste the code into the Declarations section of a form that contains two labels, one text box, two command buttons, two check boxes, and a Wave control. Press F5 and play with the buttons.

```
Option Explicit

Dim fChanged As Integer ' flag: has filename changed?

Sub Command1_Click ()
    ' make sure the filename is up to date
    SetFilename

    ' play the file
    Wave1.Action = 1
    EnableButtons
    Label2 = "Playing"
End Sub

Sub Command2_Click ()
    ' stop playing
    Wave1.Action = 4

    ' update buttons
    EnableButtons
End Sub

Sub Check1_Click ()
    Wave1.Exclusive = (Check1.Value <> 0)
End Sub

Sub Check2_Click ()
    Wave1.Loop = (Check2.Value <> 0)
End Sub

Sub EnableButtons ()
    ' set button status
    Select Case Wave1.Status
        Case 1: ' playing
            Command2.Enabled = True

        Case 4: ' stopped
            Command2.Enabled = False
    End Select
End Sub

Sub SetFilename ()
    ' if the filename has changed, tell Wave VBX
    If fChanged Then
        ' changed no more
        fChanged = False
        Wave1.Filename = Text1
    End If
End Sub
```

```

End Sub

Sub Text1_Change ()
    fChanged = True
End Sub

Sub Wave1_PlayDone ()
    EnableButtons
    Label2 = "Stopped"
End Sub

Sub Wave1_PlayLoop (LoopCount As Integer)
    Label2 = "Playing (" & LoopCount & " times)"
End Sub

Sub Form_Load ()
    Form1.BackColor = &H00C0C0C0&
    Form1.Caption = "Wave VBX Loop Sample"
    Form1.Height = 3390
    Form1.Width = 4215

    Check1.BackColor = &H00C0C0C0&
    Check1.Caption = "Exclusive Playing"
    Check1.Height = 255
    Check1.Width = 3615

    Check2.BackColor = &H00C0C0C0&
    Check2.Caption = "Repeated Playing (loop)"
    Check2.Height = 255
    Check2.Width = 3615

    Text1.Height = 285
    Text1.Text = "C:\WINDOWS\RINGIN.WAV"
    Text1.Width = 3615

    Command2.Caption = "Stop"
    Command2.Height = 615
    Command2.Width = 1815

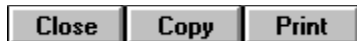
    Command1.Caption = "Play"
    Command1.Height = 615
    Command1.Width = 1815

    Label2.Alignment = 2 'Center
    Label2.BackColor = &H00C0C0C0&
    Label2.Height = 255
    Label2.Width = 3615

    Label1.BackColor = &H00C0C0C0&
    Label1.Caption = "Filename:"
    Label1.Height = 255
    Label1.Width = 1935

    fChanged = True
    EnableButtons
End Sub

```



PlayStart and PlayEnd Properties Example

In this example, the program shows how to use the PlayStart and PlayEnd properties. It plays the file shown in the edit box between the spots defined by the start and end edit boxes. To try this example, paste the code into the Declarations section of a form that contains four labels, two command buttons, one timer, four text boxes, and a Wave control. Press F5 and play with the buttons.

```
Option Explicit
```

```
Dim fChanged As Integer ' flag: has filename changed?
```

```
Sub Command1_Click ()  
    ' make sure the filename is up to date  
    SetFilename
```

```
    ' check for start field  
    If Trim(Text2) <> "" Then  
        Wave1.PlayStart = Val(Text2)  
    Else  
        Wave1.PlayStart = -1  
    End If
```

```
    ' check for end field  
    If Trim(Text3) <> "" Then  
        Wave1.PlayEnd = Val(Text3)  
    Else  
        Wave1.PlayEnd = -1  
    End If
```

```
    ' play the file  
    Wave1.Action = 1  
    EnableButtons  
End Sub
```

```
Sub Command2_Click ()  
    ' stop playing  
    Wave1.Action = 4
```

```
    ' update buttons  
    EnableButtons  
End Sub
```

```
Sub EnableButtons ()  
    Select Case Wave1.Status  
        Case 1: ' playing  
            Command2.Enabled = True  
  
        Case 2: ' paused  
            Command2.Enabled = True  
  
        Case 4: ' stopped  
            Command2.Enabled = False  
    End Select  
End Sub
```

```

Sub SetFilename ()
    ' if the filename has changed, tell Wave VBX
    If fChanged Then
        ' changed no more
        fChanged = False
        Wave1.FileName = Text4
    End If
End Sub

Sub Timer1_Timer ()
    If Wave1.Position = -1 Then
        Text1 = ""
    Else
        Text1 = Format(Wave1.Position / 1000, "0.000") & " seconds"
    End If
End Sub

Sub Text4_Change ()
    fChanged = True
End Sub

Sub Wave1_PlayDone ()
    EnableButtons
End Sub

Sub Form_Load ()
    ' set-up form, make it look nice
    Form1.BackColor = &H00C0C0C0
    Form1.BorderStyle = 3 'Fixed Double
    Form1.Caption = "Wave VBX Start/End Sample"
    Form1.Height = 3540
    Form1.Width = 4215

    Label1.BackColor = &H00C0C0C0
    Label1.Caption = "Filename:"
    Label1.Height = 255
    Label1.Width = 1935

    Label2.BackColor = &H00C0C0C0
    Label2.Caption = "Start:"
    Label2.Height = 255
    Label2.Width = 1215

    Label3.BackColor = &H00C0C0C0
    Label3.Caption = "End:"
    Label3.Height = 255
    Label3.Width = 1215

    Label4.BackColor = &H00C0C0C0
    Label4.Caption = "Position:"
    Label4.Height = 255
    Label4.Width = 1215

    Command1.Caption = "Play"
    Command1.Height = 615
    Command1.Width = 1815

```

```
Command2.Caption = "Stop"
Command2.Height = 615
Command2.Width = 1815

Timer1.Interval = 50

Text1.Height = 285
Text1.Text = ""
Text1.Width = 3615

Text2.Height = 285
Text2.Text = ""
Text2.Width = 1695

Text3.Height = 285
Text3.Text = ""
Text3.Width = 1695

Text4.Height = 285
Text4.Text = "C:\WINDOWS\RINGOUT.WAV"
Text4.Width = 3615

Wave1.Left = 3480
Wave1.Top = 0

fChanged = True
EnableButtons
End Sub
```

PlayDone Event

[See Also](#) [Example](#)

Description

Occurs when the file stops playing.

Syntax

Sub *ctlname* **PlayDone** ()

Remarks

This event fires when the current file stops playing (for whatever reason).

See Also

Properties:

[Loop](#)

Events:

[PlayLoop](#)

PlayLoop Event

[See Also](#)

[Example](#)

Description

Fires when a sound file loops (due to the [Loop](#) property).

Syntax

Sub *ctlname_PlayLoop* (*LoopCount* **As Integer**)

Remarks

This event fires when a sound file stops playing and loops back to the beginning (when the [Loop](#) is set to True). The only parameter to this routine, *LoopCount*, holds the number of times the current file has played.

See Also

Properties:

[Loop](#)

Events:

[PlayDone](#)

Registration Information

Credits

Wave was written by James Shields. Inquiries can be sent to 71231,2066 on CompuServe, or mabry@halcyon.com on Internet. Our mailing address is:

Mabry Software, Inc.
Post Office Box 31926
Seattle, WA 98103-1926

Registration

You can register this program by sending \$15 (\$20 for international orders) and your address. You can register Wave **and** its C source code by sending \$40 (\$45 for international orders). With your order, you will receive a copy of our manual documenting all of our controls.

For your convenience, an order form has been provided that you can print out directly from this help file.

E-mail Discount

You may take a \$5 discount for e-mail delivery of this package (CompuServe or Internet). If you choose this option, please note: a printed manual is not included. Be sure to include your full mailing address with your order. Sometimes (on the Internet) the package cannot be e-mailed. So, we are forced to send it through the normal mails.

CompuServe members may also take the \$5 e-mail discount by registering this package in the software registration forum (GO SWREG). Waves SWREG ID number is 4034. The source code version's ID number is 4035.

Credit Card Orders

We accept VISA and Mastercard. If you e-mail your order to us, please be sure to include your card number, expiration date, complete mailing address, and your phone number (in case we have any questions about your order).

© Copyright 1994-1995 by Mabry Software, Inc.





Wave Order Form

Use the Print Topic.. command from the File menu to print this order form.

Mail this form to: Mabry Software, Inc.
Post Office Box 31926
Seattle, WA 98103-1926
Phone: 206-634-1443
Fax: 206-632-0272
BBS: WinDev BBS 206-634-0783
CompuServe: 71231,2066
Internet: mabry@halcyon.com

Where did you get this copy of Wave?

Ship to:

Phone:

Fax:

E-Mail:

MC/VISA:

_____ exp. _____

Disk Size:

(circle one) 3½ 5¼

qty ordered _____

REGISTRATION

\$15 each, postpaid (check or money order in hard currency). Outside of North America add \$5.00 shipping.

qty ordered _____

SOURCE CODE AND REGISTRATION

\$40 each, postpaid (check or money order in hard currency). Outside of North America add \$5.00 shipping.

Getting Custom Controls Written

If you or your organization would like to have custom controls written, you can contact me at the following:

James Shields
Mabry Software, Inc.
Post Office Box 31926
Seattle, WA 98103-1926
Phone: 206-634-1443
Fax: 206-632-0272
BBS: WinDev BBS 206-634-0783
CompuServe: 71231,2066
Internet: mabry@halcyon.com

You can also contact Zane Thomas. He can be reached at:

Zane Thomas
Post Office Box 300
Indianola, WA 98342
CompuServe: 72060,3327

