

Wave Mixer ActiveX 1.0

Wave Mixer ActiveX is a 32bit OLE Control Module to mix wave audio at run-time. Wave Mixer ActiveX has following features:

- Unlimited channels *
- Set any format (bits, stereo/mono, HZ) of the output wave **
- Save output wave into file
- Set the left and right volume of the output wave ***
- Set the left and right volume of every channel
- Set start position of every channel
- Mute and unmute every channel
- Play/stop/pause/resume/seek when playing ***
- Loop playing
- Playing start position

* Mixing every channel needs CPU time. The real number of channels supported is related to your CPU speed. When on a slow CPU, I suggest you to save the mixed wave and play after.

** Format changing needs CPU time. To get the best performance, it's better to use the same format as every channel. Format changing also will lose quantity in some condition.

*** As mostly of the hardware support these features. Some sound cards or hardware don't support some features.

Wave Mixer ActiveX supports most popular development platforms including VB, VC++, VFP, VBA, Access, Delphi, etc.

Wave Mixer ActiveX is shareware. If you are using an unregistered version, please see [Registration](#) to get information about how to register.

If you have any questions, please send E-mail to jinhui@jcomsoft.com

Please visit my home page <http://www.jcomsoft.com> to get the newest version, news and more great and **FREE** ActiveX.

[Now you can register your copy online via World Wide Web.](#)

Register the Wave Mixer ActiveX

You can register Wave Mixer ActiveX 1.x for US \$40.00. After you register, you will receive a registration key that you can input by clicking the Register button in the About dialog. And you can distribute the ActiveX with your applications to your customers.

To register, please choose a registration name which must be related to your name or organization name (do not use words like Anonymous), fill and print the registration form and send check, bank draft or cash to:

JIN HUI

**Address: 501/5/285 Xin Hu Rd.
Shanghai 200436
People's Republic of China**

Note: The name on the check, bank draft and letter must be JIN HUI. Otherwise, the check or bank draft will be rejected by bank, the letter will be returned to the sender.

If you have credit cards, now you can register Wave Mixer ActiveX 1.x on my web page at <http://www.jcomsoft.com>. You will get the registration key within 72 hours via E-mail.

When I receive your mail, you will receive the registration name and registration key through E-mail.

The letters from America to China will take about ten days, and letters from Europe and Asia will take about one week. Letters from some areas such as Taiwan, Russia will take longer than two weeks.

If you want the registration key urgently, please send the letter via express mail services such as EMS, DHL or UPS. I will receive your order within three days.

If you cannot receive any information three weeks after you send the order, please contact us through E-mail jinhui@jcomsoft.com

Wave Mixer ActiveX Registration Form

Unit Price: US \$40.00

Name _____
Organization _____
Address _____
City _____
Zip code _____
Country _____
Phone (____) _____
Email _____

Preferred registration name (max 15 chars, can use space)

Any questions and comments here:

ChannelStart Property

Sets the start position of the channel

Data Type:

Long

Syntax

ChannelStart(ChannelIndex)

Remarks:

The unit of the value is a sample.

See Also:

[ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [ReadChannelWave](#), [AddChannel](#), [RemoveChannel](#)

ChannelLength Property

Gets the length of the channel

Data Type:

Long

Syntax

ChannelLength(ChannelIndex)

Remarks:

The unit of the value is a sample.

See Also:

[ChannelStart](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [ReadChannelWave](#), [AddChannel](#), [RemoveChannel](#)

ChannelLeftVolume Property

Sets or gets the left volume of the channel

Data Type:

Long

Syntax

ChannelLeftVolume(ChannelIndex)

Remarks:

The value is from 0 to 100. 0 means silence. 100 means full volume.

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [ReadChannelWave](#), [AddChannel](#), [RemoveChannel](#)

ChannelRightVolume Property

Sets or gets the right volume of the channel

Data Type:

Long

Syntax

ChannelRightVolume(ChannelIndex)

Remarks:

The value is from 0 to 100. 0 means silence. 100 means full volume.

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [ReadChannelWave](#), [AddChannel](#), [RemoveChannel](#)

ChannelMute Property

Mutes or unmutes the channel

Data Type:

Boolean

Syntax

ChannelMute(ChannelIndex)

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [ReadChannelWave](#), [AddChannel](#), [RemoveChannel](#)

ChannelStereo Property

Gets if the channel is stereo or mono.

Data Type:

Boolean

Syntax

ChannelStereo(ChannelIndex)

Remarks:

The value only can be read.

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [ReadChannelWave](#), [AddChannel](#), [RemoveChannel](#)

ChannelSamplesPerSec Property

Gets the number of the samples per second in the channel.

Data Type:

Long

Syntax

ChannelSamplesPerSec(ChannelIndex)

Remarks:

The value only can be read. The common values are 8000 (8.0kHz), 11025 (11.025 kHz), 22050 (22.05 kHz) and 44100 (44.1 kHz).

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelBitsPerSample](#), [ReadChannelWave](#), [AddChannel](#), [RemoveChannel](#)

ChannelBitsPerSample Property

Gets the bits of every sample in in the channel.

Data Type:

Long

Syntax

ChannelBitsPerSample(ChannelIndex)

Remarks:

The value only can be read. The common values are 8 (8 bits audio) and 16 (16 bits audio).

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ReadChannelWave](#), [AddChannel](#), [RemoveChannel](#)

ReadChannelWave Method

Reads in the wave file into the Channel.

Syntax

```
ReadChannelWave( ChannelIndex, FileName )
```

Return value

Boolean

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [AddChannel](#), [RemoveChannel](#)

AddChannel Method

Adds a new channel

Syntax

AddChannel

Return value

Long

Remark

Returns the index of the new channel

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [RemoveChannel](#)

RemoveChannel Method

Remove a channel

Syntax

RemoveChannel (ChannelIndex)

Return value

Boolean

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [AddChannel](#)

ChannelCount Property

Gets the number of current channels.

Data Type:

Long

See Also:

[ChannelStart](#), [ChannelLength](#), [ChannelLeftVolume](#), [ChannelRightVolume](#), [ChannelMute](#), [ChannelStereo](#), [ChannelSamplesPerSec](#), [ChannelBitsPerSample](#), [AddChannel](#)

LeftVolume Property

Sets or gets the left volume

Data Type:

Long

Remarks:

The value is from 0 to 100. 0 means silence. 100 means full volume. Some hardware only recognizes the left volume or uses the average value of the left volume and the right volume.

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#)

RightVolume Property

Sets or gets the right volume

Data Type:

Long

Remarks:

The value is from 0 to 100. 0 means silence. 100 means full volume. Some hardware only recognizes the left volume or uses the average value of the left volume and the right volume.

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#)

StartPos Property

Sets or gets the start position when playing

Data Type:

Long

Remarks:

The unit of the value is a sample.

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#)

IgnoreChannelVolume Property

Sets or gets if the volume of the channel will be used in mixing.

Data Type:

Boolean

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#)

Stereo Property

Sets or gets if the output wave is stereo or mono.

Data Type:

Boolean

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#)

SamplesPerSec Property

Sets or gets the number of the samples per second in the output wave.

Data Type:

Long

Remarks:

The common values are 8000 (8.0kHz), 11025 (11.025 kHz), 22050 (22.05 kHz) and 44100 (44.1 kHz).

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#)

BitsPerSample Property

Sets or gets the bits of every sample in the output wave.

Data Type:

Long

Remarks:

The common values are 8 (8 bits audio) and 16 (16 bits audio).

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#)

Play Method

Plays the mixed wave.

Syntax

Play

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#), [Play](#), [Stop](#), [Pause](#), [Resume](#), [Seek](#), [Loop](#)

Stop Method

Stops the mixed wave.

Syntax

Stop

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#), [Play](#), [Stop](#), [Pause](#), [Resume](#), [Seek](#), [Loop](#)

Pause Method

Pauses the playing mixed wave.

Syntax

Pause

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#), [Play](#), [Stop](#), [Pause](#), [Resume](#), [Seek](#), [Loop](#)

Resume Method

Resumes the paused mixed wave.

Syntax

Resume

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#), [Play](#), [Stop](#), [Pause](#), [Resume](#), [Seek](#), [Loop](#)

Seek Method

Seeks the playing position when playing the mixed wave.

Syntax

Seek (NewPosition)

Remark

The new position is in unit of a sample.

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#), [Play](#), [Stop](#), [Pause](#), [Resume](#), [Seek](#), [Loop](#)

Loop Property

Sets or gets if the playing is loop or not.

Data Type:

Boolean

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#), [Play](#), [Stop](#), [Pause](#), [Resume](#), [Seek](#), [Loop](#)

SaveMixedWave Method

Saves the mixed wave into a file.

Syntax

SaveMixedWave (FileName)

Return Value

Boolean

See Also:

[StartPos](#), [LeftVolume](#), [RightVolume](#), [IgnoreChannelVolume](#), [Stereo](#), [SamplesPerSec](#), [BitsPerSample](#)

PlayFinished Event

Fires when the playing is finished.

See Also:

[Play](#), [Stop](#), [Pause](#), [Resume](#), [Seek](#), [Loop](#)

