i3DBuffer supports the following properties:

ConeOrientationX	X orientation of the sound projection cone
ConeOrientationY	Y orientation of the sound projection cone
ConeOrientationZ	Z orientation of the sound projection cone
ConeOutsideVolume	Outside cone volume
InsideConeAngle	The angle of the inside sound projection cone
MaxDistance	Maximum distance
MinDistance	Minimum distance
Mode	3D sound processing mode
OutsideConeAngle	The angle of the outside sound projection cone
PositionX	X Position of the 3D speaker
PositionY	Y Position of the 3D speaker
PositionZ	Z Position of the 3D speaker
VelocityX	X Velocity of the 3D speaker
VelocityY	Y Velocity of the 3D speaker
VelocityZ	Z Velocity of the 3D speaker

The current X orientation of the sound projection cone.

Syntax

i3DBuffer.**ConeOrientationX** [= *ConeOrientationX*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vConeOrientation member of the DS3DBUFFER structure

See Also

None

The current Y orientation of the sound projection cone.

Syntax

i3DBuffer.**ConeOrientationY** [= *ConeOrientationY*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vConeOrientation member of the DS3DBUFFER structure

See Also

None

The current Z orientation of the sound projection cone.

Syntax

i3DBuffer.**ConeOrientationZ** [= *ConeOrientationZ*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vConeOrientation member of the DS3DBUFFER structure

See Also

None

The cone outside volume.

Syntax

i3DBuffer.**ConeOutsideVolume** [= *ConeOutsideVolume*]

Type

Long

Remarks

None

Direct X Compatibility

This property is the equivalent of the lConeOutsideVolume   member of the DS3DBUFFER structure

See Also

None

The angle of the inside sound projection cone.

Syntax

i3DBuffer.**InsideConeAngle** [= *InsideConeAngle*]

Type

Long

Remarks

None

Direct X Compatibility

This property is the equivalent of the dwInsideConeAngle   member of the DS3DBUFFER structure

See Also

None

The maximum distance.

Syntax

i3DBuffer.**MaxDistance** [= *MaxDistance*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the flMaxDistance member of the DS3DBUFFER structure

See Also

None

The minimum distance.

Syntax

i3DBuffer.**MinDistance** [= *MinDistance*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the flMinDistance   member of the DS3DBUFFER structure

See Also

None

The 3D sound processing mode.

Syntax

i3DBuffer.**Mode** [= *Mode*]

Type

Long

Remarks

Set to one of :

DS3DMODE_DISABLE

DS3DMODE_HEADRELATIVE

DS3DMODE_NORMAL

Direct X Compatibility

This property is the equivalent of the dwMode member of the DS3DBUFFER structure

See Also

None

The angle of the outside sound projection cone.

Syntax

i3DBuffer.**OutsideConeAngle** [= *OutsideConeAngle*]

Type

Long

Remarks

None

Direct X Compatibility

This property is the equivalent of the dwOutsideConeAngle   member of the DS3DBUFFER structure

See Also

None

The X position of the 3D sound buffer.

Syntax

i3DBuffer.**PositionX** [= *PositionX*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vPosition member of the DS3DBUFFER structure

See Also

None

The Y position of the 3D sound buffer.

Syntax

i3DBuffer.**PositionY** [= *PositionY*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vPosition member of the DS3DBUFFER structure

See Also

None

The X position of the 3D sound buffer.

Syntax

i3DBuffer.**PositionZ** [= *PositionZ*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vPosition member of the DS3DBUFFER structure

See Also

None

The X velocity of the 3D sound buffer.

Syntax

i3DBuffer.**VelocityX** [= *VelocityX*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vVelocity member of the DS3DBUFFER structure

See Also

None

The Y velocity of the 3D sound buffer.

Syntax

i3DBuffer.**VelocityY** [= *VelocityY*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vVelocity member of the DS3DBUFFER structure

See Also

None

The Z velocity of the 3D sound buffer.

Syntax

i3DBuffer.**VelocityZ** [= *VelocityZ*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vVelocity member of the DS3DBUFFER structure

See Also

None

i3DListener supports the following properties

DistanceFactor	Distance factor
DopplerFactor	Doppler factor
OrientFrontX	Front orientation in X direction
OrientFrontY	Front orientation in Y direction
OrientFrontZ	Front orientation in Z direction
OrientTopX	Top orientation in X direction
OrientTopY	Top orientation in Y direction
OrientTopZ	Top orientation in Z direction
PositionX	X Position
PositionY	Y Position
PositionZ	Z Position
RolloffFactor	Roll off factor
VelocityX	X Velocity
VelocityY	Y Velocity
VelocityZ	Z Velocity

The current distance

Syntax

i3DListener.**DistanceFactor** [= *DistanceFactor*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the flDistanceFactor member of the DS3DLISTENER structure

See Also

None

The current Doppler factor.

Syntax

i3DListener.**DopplerFactor** [= *DopplerFactor*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the flDopplerFactor member of the DS3DLISTENER structure

See Also

None

The listener's front orientation in the X direction.

Syntax

i3DListener.**OrientFrontX** [= *OrientFrontX*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vOrientFront member of the DS3DLISTENER structure

See Also

None

The listener's front orientation in the Y direction

Syntax

i3DListener.**OrientFrontY** [= *OrientFrontY*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vOrientFront member of the DS3DLISTENER structure

See Also

None

The listener's front orientation in the Z direction

Syntax

i3DListener.**OrientFrontZ** [= *OrientFrontZ*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vOrientFront member of the DS3DLISTENER structure

See Also

None

The listener's top orientation in the X direction.

Syntax

i3DListener.**OrientTopX** [= *OrientTopX*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vOrientTop member of the DS3DLISTENER structure

See Also

None

The listener's top orientation in the Y direction.

Syntax

i3DListener.**OrientTopY** [= *OrientTopY*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vOrientTop member of the DS3DLISTENER structure

See Also

None

The listener's top orientation in the Z direction.

Syntax

i3DListener.**OrientTopZ** [= *OrientTopZ*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vOrientTop member of the DS3DLISTENER structure

See Also

None

The listener's position in the X direction.

Syntax

i3DListener.**PositionX** [= *PositionX*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vPosition member of the DS3DLISTENER structure

See Also

None

The listener's position in the Y direction.

Syntax

i3DListener.**PositionY** [= *PositionY*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vPosition member of the DS3DLISTENER structure

See Also

None

The listener's position in the Z direction.

Syntax

i3DListener.**PositionZ** [= *PositionZ*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vPosition member of the DS3DLISTENER structure

See Also

None

The rolloff factor

Syntax

i3DListener.**RolloffFactor** [= *RolloffFactor*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the flRolloffFactor member of the DS3DLISTENER structure

See Also

None

The listener's velocity in the X direction.

Syntax

i3DListener.**VelocityX** [= *VelocityX*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vVelocity member of the DS3DLISTENER structure

See Also

None

The listener's velocity in the Y direction.

Syntax

i3DListener.**VelocityY** [= *VelocityY*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vVelocity member of the DS3DLISTENER structure

See Also

None

The listener's velocity in the Z direction.

Syntax

i3DListener.**VelocityZ** [= *VelocityZ*]

Type

Float

Remarks

None

Direct X Compatibility

This property is the equivalent of the vVelocity member of the DS3DLISTENER structure

See Also

None

iBCaps supports the following properties:

BufferBytes	Size of buffer in bytes
Flags	Specifies the buffer capabilities
PlayCpuOverhead	Specifies the processing overhead needed to mix a sound buffer
UnlockTransferRate	Specifies the rate, in kilobytes per second, that data is transferred to the buffer memory

Size of this buffer, in bytes.

Syntax

iBCaps.**BufferBytes** [= *BufferBytes*]

Type

Long

Remarks

None

Direct X Compatibility

This property is the equivalent of the dwBufferBytes member of the DSBCAPS structure

See Also

None

Buffer capabilities

Syntax

iBCaps.**Flags** [= *Flags*]

Type

Long

Remarks

Flags that specify buffer-object capabilities.

DSBCAPS_CTRL3D
DSBCAPS_CTRLFREQUENCY
DSBCAPS_CTRLPAN
DSBCAPS_CTRLVOLUME
DSBCAPS_GETCURRENTPOSITION2
DSBCAPS_GLOBALFOCUS
DSBCAPS_LOCHARDWARE
DSBCAPS_LOCSOFTWARE
DSBCAPS_PRIMARYBUFFER
DSBCAPS_STATIC
DSBCAPS_STICKYFOCUS

Direct X Compatibility

This property is the equivalent of the dwFlags member of the DSBCAPS structure

See Also

None

Specifies the processing overhead as a percentage of main processing cycles needed to mix this sound buffer.

Syntax

iBCaps.**PlayCpuOverhead** [= *PlayCpuOverhead*]

Type

Long

Remarks

None

Direct X Compatibility

This property is the equivalent of the dwPlayCpuOverhead member of the DSBCAPS structure

See Also

None

Specifies the rate, in kilobytes per second, that data is transferred to the buffer memory when IDirectSoundBuffer::Unlock is called

Syntax

iBCaps.**UnlockTransferRate** [= *UnlockTransferRate*]

Type

Long

Remarks

None

Direct X Compatibility

This property is the equivalent of the dwUnlockTransferRate member of the DSBCAPS structure

See Also

None

iBufferDesc supports the following methods:

Reset	Resets all iBufferDesc properties to NULL

Resets all iBufferDesc properties to NULL

Syntax

iBufferDesc**.Reset ( )**

Type

void

Parameters

None

Remarks

Use this method to reset all the properties of an iBufferDesc object. This is useful if you need to reuse the object.

Direct X Compatibility

This method has no Direct X equivalent

See Also

None

iBufferDesc supports the following properties:

BufferBytes	Size of the buffer, in bytes
Flags	Capabilities of the sound buffer
WfxFormat	An iWaveFormatEx object

Size of the new buffer, in bytes.

Syntax

iBufferDesc**.BufferBytes** [= *BufferBytes*]

Type

Long

Remarks

This value must be 0 when creating primary buffers

Direct X Compatibility

This property is the equivalent of the dwBufferBytes member of the DSBUFFERDESC structure

See Also

None

Identifies the capabilities to include when creating a new iSoundBuffer object.

Syntax

iBufferDesc.**Flags** [= *Flags*]

Type

Long

Remarks

Specify one or more of the following:

DSBCAPS_CTRL3D

DSBCAPS_CTRLALL

DSBCAPS_CTRLDEFAULT

DSBCAPS_CTRLFREQUENCY

DSBCAPS_CTRLPAN

DSBCAPS_CTRLVOLUME

DSBCAPS_GETCURRENTPOSITION2

DSBCAPS_GLOBALFOCUS

DSBCAPS_LOCHARDWARE

DSBCAPS_LOCSOFTWARE

DSBCAPS_PRIMARYBUFFER

DSBCAPS_STATIC

DSBCAPS_STICKYFOCUS

Direct X Compatibility

This property is the equivalent of the dwFlags member of the DSBUFFERDESC structure

See Also

None

An iWaveFormatEx object.

Syntax

iBufferDesc.**WfxFormat** [= *WfxFormat*]

Type

iWaveFormatEx Object

Remarks

This value must be NULL for primary buffers. The application can use the Format method to set the format of the primary buffer.

For secondary buffers use the CreateWfxFormat method to create an iWaveFormatEx object. Set its properties as required and then set this property to the iWaveFormatEx object.

Direct X Compatibility

This property is the equivalent of the lpwfxFormat member of the DSBUFFERDESC structure

See Also

iWaveFormatEx object

iCaps supports the following properties.

Flags	Device Capabilities
FreeHw3DAllBuffers	3D capability of the Hardware
FreeHw3DStaticBuffers	3D capability of the Hardware
FreeHw3DStreamingBuffers	3D capability of the Hardware
FreeHwMemBytes	Size, in bytes, of free memory on the sound card.
FreeHwMixingAllBuffers	Free hardware mixing capability
FreeHwMixingStaticBuffers	Free hardware mixing capability
FreeHwMixingStreamingBuffers	Free hardware mixing capability
MaxContigFreeHwMemBytes	Size, in bytes, of the largest contiguous block of free memory on the sound card
MaxHw3DAllBuffers	3D capability of the Hardware
MaxHw3DStaticBuffers	3D capability of the Hardware
MaxHw3DStreamingBuffers	3D capability of the Hardware
MaxHwMixingAllBuffers	Total number of buffers that can be mixed in hardware
MaxHwMixingStaticBuffers	Maximum number of static sound buffers
MaxHwMixingStreamingBuffers	Maximum number of streaming sound buffers
MaxSecondarySampleRate	Maximum sample rate that is supported by this device's hardware secondary sound buffers.
MinSecondarySampleRate	Minimum sample rate that is supported by this device's hardware secondary sound buffers.
PlayCpuOverheadSwBuffers	Processing overhead, as a percentage of the central processing unit, needed to mix software buffers
PrimaryBuffers	Number of primary buffers supported.
TotalHwMemBytes	Size, in bytes, of the amount of memory on the sound card that stores static sound buffers.
UnlockTransferRateHwBuffers	Rate, in kilobytes per second, at which data can be transferred to hardware static sound buffers

Specifies device capabilities.

Syntax

iCaps.**Flags** [= *Flags*]

Type

Long

Remarks

Can be one or more of the following:

DSCAPS_CERTIFIED

DSCAPS_CONTINUOUSRATE

DSCAPS_EMULDRIVER

DSCAPS_PRIMARY16BIT

DSCAPS_PRIMARY8BIT

DSCAPS_PRIMARYMONO

DSCAPS_PRIMARYSTEREO

DSCAPS_SECONDARY16BIT

DSCAPS_SECONDARY8BIT

DSCAPS_SECONDARYMONO

DSCAPS_SECONDARYSTEREO

Direct X Compatibility

This property is equivalent to the dwFlags member of the DSCAPS structure

See Also

None

Description of the free, or unallocated, hardware 3D positional capabilities of the device. This will always be 0 for this release.

Syntax

iCaps.**FreeHw3DAllBuffers** [= *FreeHw3DAllBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwFreeHw3DAllBuffers member of the DSCAPS structure

See Also

None

Description of the free, or unallocated, hardware 3D positional capabilities of the device. This will always be 0 for this release.

Syntax

iCaps.**FreeHw3DStaticBuffers** [= *FreeHw3DStaticBuffers*]

Type

Long

Remarks

A static buffer is a section of memory that contains a complete sound. These buffers are convenient because the entire sound can be written once to the buffer.

Direct X Compatibility

This property is equivalent to the dwFreeHw3DStaticBuffers member of the DSCAPS structure

See Also

None

Description of the free, or unallocated, hardware 3D positional capabilities of the device. This will always be 0 for this release.

Syntax

iCaps.**FreeHw3DStreamingBuffers** [= *FreeHw3DStreamingBuffers*]

Type

Long

Remarks

A streaming buffer is a small sound buffer that can play lengthy sounds because the application dynamically loads audio data into the buffer as it plays. For example, an application could use a buffer that can hold 3 seconds of audio data to play a 2-minute sound. A streaming buffer requires much less memory than a static buffer.

Direct X Compatibility

This property is equivalent to the dwFreeHw3DStreamingBuffers member of the DSCAPS structure

See Also

Size, in bytes, of the free memory on the sound card.

Syntax

iCaps.**FreeHwMemBytes** [= *FreeHwMemBytes*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwFreeHwMemBytes member of the DSCAPS structure

See Also

None

Free, or unallocated, hardware mixing capabilities of the device.

Syntax

iCaps.**FreeHwMixingAllBuffers** [= *FreeHwMixingAllBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwFreeHwMixingAllBuffers member of the DSCAPS structure

See Also

None

Free, or unallocated, hardware mixing capabilities of the device.

Syntax

iCaps.**FreeHwMixingStaticBuffers** [= *FreeHwMixingStaticBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwFreeHwMixingStaticBuffers member of the DSCAPS structure

See Also

None

Free, or unallocated, hardware mixing capabilities of the device.

Syntax

iCaps.**FreeHwMixingStreamingBuffers** [= *FreeHwMixingStreamingBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwFreeHwMixingStreamingBuffers member of the DSCAPS structure

See Also

Size, in bytes, of the largest contiguous block of free memory on the sound card.

Syntax

iCaps.**MaxContigFreeHwMemBytes** [= *MaxContigFreeHwMemBytes*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwMaxContigFreeHwMemBytes member of the DSCAPS structure

See Also

None

Hardware 3D positional capabilities of the device. This will always be 0 for this release.

Syntax

iCaps.**MaxHw3DAllBuffers** [= *MaxHw3DAllBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwMaxHw3DAllBuffers member of the DSCAPS structure

See Also

None

Hardware 3D positional capabilities of the device. This will always be 0 for this release.

Syntax

iCaps.**MaxHw3DStaticBuffers** [= *MaxHw3DStaticBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwMaxHw3DStaticBuffers member of the DSCAPS structure

See Also

None

Hardware 3D positional capabilities of the device. This will always be 0 for this release.

Syntax

iCaps.**MaxHw3DStreamingBuffers** [= *MaxHw3DStreamingBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwMaxHw3DStreamingBuffers member of the DSCAPS structure

See Also

None

Specifies the total number of buffers that can be mixed in hardware.

Syntax

iCaps.**MaxHwMixingAllBuffers** [= *MaxHwMixingAllBuffers*]

Type

Long

Remarks

This member can be less than the sum of dwMaxHwMixingStaticBuffers and dwMaxHwMixingStreamingBuffers. Resource trade-offs frequently occur.

Direct X Compatibility

This property is equivalent to the dwMaxHwMixingAllBuffers member of the DSCAPS structure

See Also

None

Specifies the maximum number of static sound buffers.

Syntax

iCaps.**MaxHwMixingStaticBuffers** [= *MaxHwMixingStaticBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwMaxHwMixingStaticBuffers member of the DSCAPS structure

See Also

None

Specifies the maximum number of streaming sound buffers.

Syntax

iCaps.**MaxHwMixingStreamingBuffers** [= *MaxHwMixingStreamingBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwMaxHwMixingStreamingBuffers member of the DSCAPS structure

See Also

None

Maximum sample rate specifications that are supported by this device's hardware secondary sound buffers.

Syntax

iCaps.**MaxSecondarySampleRate** [= *MaxSecondarySampleRate*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwMaxSecondarySampleRate member of the DSCAPS structure

See Also

None

Minimum sample rate specifications that are supported by this device's hardware secondary sound buffers.

Syntax

iCaps.**MinSecondarySampleRate** [= *MinSecondarySampleRate*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwMinSecondarySampleRate member of the DSCAPS structure

See Also

None

The processing overhead, as a percentage of the central processing unit, needed to mix software buffers.

Syntax

iCaps.**PlayCpuOverheadSwBuffers** [= *PlayCpuOverheadSwBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwPlayCpuOverheadSwBuffers member of the DSCAPS structure

See Also

None

Number of primary buffers supported. This value will always be 1 for this release.

Syntax

iCaps.**PrimaryBuffers** [= *PrimaryBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwPrimaryBuffers member of the DSCAPS structure

See Also

None

Size, in bytes, of the amount of memory on the sound card that stores static sound buffers.

Syntax

iCaps.**TotalHwMemBytes** [= *TotalHwMemBytes*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwTotalHwMemBytes member of the DSCAPS structure

See Also

None

The rate, in kilobytes per second, at which data can be transferred to hardware static sound buffers.

Syntax

iCaps.**UnlockTransferRateHwBuffers** [= *UnlockTransferRateHwBuffers*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the dwUnlockTransferRateHwBuffers member of the DSCAPS structure

See Also

None

iDevice supports the following properties:

DriverDescription	Description of the device driver
DriverName	Name of the device driver

Description of the sound device driver

Syntax

iDevice**.DriverDescription** [= *DriverDescription*]

Type

String

Remarks

Provides a description of the sound device driver

Direct X Compatibility

This property is equivalent to the lpstrDescription parameter of the DSEnumCallback ( ) function

See Also

None

Name of the sound device driver

Syntax

iDevice**.DriverName** [= *DriverName*]

Type

String

Remarks

Provides the of the sound device driver

Direct X Compatibility

This property is equivalent to the lpstrModule parameter in DSEnumCallback ( ) function

See Also

None

iSound OCX Features :

- ActiveX component interface to DirectSound3
- Simplifies DirectSound Programming
- Allows access to DirectSound from any OLE container - VisualBasic, Delphi, Visual C++ and many more

CompleteControl ™

What is iSound

More About iSound

iSound Reference

System Requirements

iSound Objects

Sample Applications

Examples

Contact Information

How to contact us

Obtaining Technical Support

Registration

Registering your purchase

Licensing and Distribution

License Agreement

Distributing iSound

Un-Installing iSound

iSound is a 32 bit ActiveX Control wrapper around the Microsoft® DirectSound API. DirectSound is a technology that allows you to create powerful audio enabled applications for Windows.

If you are familiar with DirectX you will know it is very difficult to program from environments such as Visual Basic. We have created iSound to provide easy access to DirectSound from virtually any programming environment.

iSound is designed to give you complete access to the DirectSound API. You should to be familiar with DirectSound before using iSound OCX. This on-line help does provide you with a complete iSound reference however DirectSound programming concepts are not covered here in depth – please consult the DirectSound documentation from Microsoft.

iSound is just one of many software components from our CompleteControl range of products, including other DirectX components - i3D, iDraw, iPlay and iInput.

iSound is supported under the following 32 bit Microsoft® Windows operating systems on Intel® architecture machines.

Windows 95

Windows NT Server and Workstation 3.51

Windows NT Server and Workstation 4.0

and future versions of these operating systems.

Note:

The current version of iSound is not supported on beta versions of the above operating systems, Win32s or on non-Intel architecture implementations of Microsoft Windows.

For information about future versions and updates please register with Imagine IT Ltd

To register your copy of iSound please print out this page and send it to us after completing the details. Alternatively you can also email the required information to us at: **registration@imagineit.co.uk**

Registered users automatically receive information about updates, new releases and future products from Imagine IT and are eligible for technical support.

Product: iSound OCX v 2.0

Last Name:

————————————————————

First Name:

————————————————————---

Company Name:

————————————————————----

Address:

————————————————————----

————————————————————----

————————————————————----

Telephone Number:

————————————————————----

————————————————————----

Fax Number:

————————————————————---

————————————————————---

Email Address:

————————————————————----

————————————————————----

Web URL:

————————————————————---

————————————————————---

Date of Purchase:

————————————————————----

Purchased From:

————————————————————----

please send to:

Imagine IT Limited, 3rd Floor, Hygeia Building, 66 College Road, Harrow HA1 1FD, United Kingdom

or email to:

registration@imagineit.co.uk

iSound is designed and developed by Imagine IT Limited.

Imagine IT specialises in object technologies and component software development.

Apart from creating great components we also help customers develop their own line-of-business objects and applications.

Contact us for more information about our products and services :

Write to us at:

Imagine IT Limited
3rd Floor,
Hygeia Building
66 College Road
Harrow, HA1 1FD
United Kingdom

Telephone:

from within the United Kingdom:	0181 324 1240
from other countries:	international access code + 44 + 181 324 1240

Fax:

from within the United Kingdom:	0181 324 1752
from other countries:	international access code + 44 + 181 324 1752

Email:

for general inquiries:	info@imagineit.co.uk
for technical support inquiries:	suport@imagineit.co.uk
for registration:	registration@imagineit.co.uk

World Wide Web:

find us at:	www.imagineit.co.uk

For the fastest possible response please direct all technical support enquiries via email to `support@imagineit.co.uk`.

Or fax your enquiry marked for the attention of “Technical Support”.

If you do not have access to email or fax then please call us at the above telephone number and ask for “Technical Support”

NOTE: We can only provide technical support to registered users.

IMPORTANT

THE USE OF THIS SOFTWARE IS SUBJECT TO THE TERMS OF THE LICENSE AGREEMENT PRINTED BELOW. PLEASE READ THE LICENSE AGREEMENT CAREFULLY. IF YOU DO NOT AGREE WITH ALL OF THE TERMS, THEN YOU SHOULD NOT USE THE SOFTWARE IN ANY WAY WHATSOEVER INCLUDING COPYING OR DISTRIBUTING THE SOFTWARE OR ANY PART OF THE SOFTWARE TO ANY THIRD PARTY. IF YOU USE THE SOFTWARE IN ANY WAY WHATSOEVER IT WILL BE DEEMED TO INDICATE YOUR ACCEPTANCE OF ALL THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT.

THE AGREEMENT

THIS DOCUMENT IS A LEGAL AGREEMENT BETWEEN IMAGINE IT LIMITED, UNITED KINGDOM, HEREAFTER REFERRED TO AS "IMAGINE IT" AND THE USER OF THE SOFTWARE HEREIN REFERRED TO AS "LICENSEE"

DEFINITIONS

"SOFTWARE" MEANS THE SOFTWARE KNOWN AS "iSound OCX" AND ALL ITS COMPONENTS, PARTS AND DOCUMENTATION, WHICH HAVE BEEN DEVELOPED BY IMAGINE IT

"END USER APPLICATION" MEANS ANY APPLICATION DEVELOPED WITH THE HELP OF OR BY USING THE SOFTWARE

PURPOSE

THE PURPOSE OF THIS AGREEMENT IS TO DEFINE THE RELATIONSHIP BETWEEN IMAGINE IT AND LICENSEE, THE TERMS, CONDITIONS, AND LIMITATIONS OF USE, THE RIGHTS OF BOTH PARTIES AND THE LIMITATION OF LIABILITY ARISING FROM THE USE OF THE SOFTWARE

EFFECTIVE DATE

THIS AGREEMENT IS DEEMED TO HAVE BEEN MADE EFFECTIVE ON THE FIRST DATE AT WHICH LICENSEE OPENS THE PACKAGE CONTAINING THE SOFTWARE AND ITS ASSOCIATED DOCUMENTATION, OR IF THE SOFTWARE IS DISTRIBUTED ELECTRONICALLY, THE FIRST DATE ON WHICH THE LICENSEE OBTAINS THE SOFTWARE

GRANT OF LIMITED LICENSE

I. EVALUATION LICENSE

IF THE LICENSEE HAS NOT PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN IMAGINE IT GRANTS THE LICENSEE A NON EXCLUSIVE, NON TRANSFERABLE, PERSONAL LICENSE TO USE THE SOFTWARE AND ITS DOCUMENTATION FOR EVALUATION PURPOSES ONLY AND FOR NO OTHER PURPOSE. THE LICENSE DOES NOT PERMIT THE DEVELOPMENT AND DISTRIBUTION OF ANY END USER APPLICATIONS USING THE SOFTWARE

II. FULL DEVELOPMENT LICENSE

IF THE LICENSEE HAS PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN IMAGINE IT GRANTS THE LICENSEE A NON EXCLUSIVE, NON TRANSFERABLE, PERSONAL LICENSE TO USE THE SOFTWARE AND ITS DOCUMENTATION FOR THE PURPOSE OF DEVELOPING AND DISTRIBUTING ANY NUMBER OF END USER APPLICATIONS AND TO COPY AND DISTRIBUTE ANY PARTS OF THE SOFTWARE DEFINED BELOW UNDER REDISTRIBUTABLE COMPONENTS TOGETHER WITH AND AS PART OF THE END USER APPLICATION, SUBJECT TO ALL THE TERMS AND CONDITIONS AND RESTRICTIONS IN THIS AGREEMENT.

RESTRICTIONS

A USER OF ANY END USER APPLICATIONS MAY NOT FURTHER USE PARTS OF THE SOFTWARE FOR SOFTWARE DEVELOPMENT, COPYING OR DISTRIBUTION. THE LICENSEE MUST ENFORCE THIS IN A SEPARATE AGREEMENT WITH THE USER OF ANY END USER APPLICATIONS.

THE LICENSEE MAY USE ONLY ONE COPY OF THE SOFTWARE AT ANY TIME ON ONE IBM PC OR COMPATIBLE SYSTEM.

ADDITIONAL PRODUCT KEYS FOR THE SOFTWARE MUST BE PURCHASED IF IT IS REQUIRED TO BE USED IN A MULTI USER NETWORKED ENVIRONMENT IN QUANTITIES OF ONE KEY FOR EACH PERSON HAVING ACCESS TO AND USING THE SOFTWARE.

THE LICENSEE MAY MAKE A SINGLE BACKUP COPY OF THE SOFTWARE.

THE LICENSEE MAY NOT USE, COPY, MODIFY, REVERSE ENGINEER, DISASSEMBLE, SELL, TRANSFER, HIRE, LEND OR OTHERWISE DISTRIBUTE THE SOFTWARE OR ANY OF ITS DOCUMENTATION OR COMPONENTS IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS AGREEMENT.

THE END USER APPLICATION MUST NOT BE ANOTHER DEVELOPMENT TOOL DESIGNED TO BE USED FOR CREATING OTHER END USER APPLICATIONS USING THE SOFTWARE. THE LICENSEE MUST ENFORCE THIS IN A SEPARATE AGREEMENT WITH THE USER OF THE END USER APPLICATION.

THE LICENSEE HAS NO RIGHTS TO THE ORIGINAL SOURCE CODE OF THE SOFTWARE WHATSOEVER.

THE LICENSEE MUST INFORM IMAGINE IT OF THE NATURE OF EACH AND EVERY NEW END USER APPLICATION IT DEVELOPS WITH THE USE OF THE SOFTWARE.

THE LICENSEE SHALL COMPLY WITH ALL LOCAL AND INTERNATIONAL LAWS AND EXPORT / IMPORT REGULATIONS WHEN DISTRIBUTING PARTS OF THIS SOFTWARE AS PROVIDED FOR BELOW ALONG WITH ANY END USER

APPLICATIONS

REDISTRIBUTABLE COMPONENTS

THE SOFTWARE, BY ITS NATURE, INCLUDES COMPONENTS WHICH ARE REQUIRED TO BE DISTRIBUTED WITH ANY END USER APPLICATIONS IN ORDER FOR THAT END USER APPLICATION TO FUNCTION.

I. EVALUATION LICENSE

IF THE LICENSEE HAS NOT PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN THERE ARE NO DISTRIBUTABLE COMPONENTS. THE LICENSEE SHALL NOT DISTRIBUTE ANY COMPONENTS OF THE SOFTWARE

II. FULL DEVELOPMENT LICENSE

IF THE LICENSEE HAS PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THENTHE FOLLOWING FILES DEFINE THE COMPONENTS WHICH THE LICENSEE MAY DISTRIBUTE PROVIDED THAT THEY ARE DISTRIBUTED AS PART OF THE END USER APPLICATION AND PROVIDED THAT THEY ARE DISTRIBUTED WITHOUT ANY MODIFICATION AND ARE COPIED FROM THE ORIGINAL DISKS OR CDS. ALL FILES LISTED BELOW MUST BE DISTRIBUTED WITH THE END USER APPLICATION.

1. **iSound.OCX**
2. **iSound.RTL**

NO OTHER FILES SUPPLIED WITH THE SOFTWARE MAY BE DISTRIBUTED WITH THE END USER APPLICATION.

THE SOFTWARE BY ITS NATURE ALSO REQUIRES SOME FILES WHICH ARE DISTRIBUTED BY MICROSOFT(R) TO BE PRESENT ON THE END USER'S COMPUTER FOR THE END USER APPLICATION TO FUNCTION. THIS LICENSE AGREEMENT DOES NOT IN ANY WAY PROVIDE FOR LICENSEE TO DISTRIBUTE THESE COMPONENTS WITH THE END USER AGREEMENT. THE LICENSEE MUST VERIFY THEIR RIGHTS TO DISTRIBUTE THESE COMPONENTS, SEPARATELY WITH MICROSOFT(R) BEFORE DISTRIBUTING THESE COMPONENTS WITH THE END USER APPLICATION

1. **MFC42.DLL**
2. **OLEPRO32.DLL**
3. **REGSVR32.EXE**

TECHNICAL SUPPORT

ONLY THOSE LICENSEES WHO HAVE PURCHASED A FULL PRODUCT KEY AND REGISTERED WITH IMAGINE IT BY FULLY COMPLETING AND RETURNING ALL THE INFORMATION ASKED FOR IN THE REGISTRATION FORM OR RELEVANT SECTION IN THE ON-LINE HELP DOCUMENTATION ARE QUALIFIED FOR RECEIVING TECHNICAL SUPPORT HELP ON THE SOFTWARE.

TECHNICAL SUPPORT IS RESTRICTED TO ANSWERING QUESTIONS ABOUT THE USE OF THE SOFTWARE OR ERRORS IN THE SOFTWARE IF ANY.

TECHNICAL SUPPORT DOES NOT INCLUDE DEBUGGING AND OTHER PROBLEM SOLVING TASKS ASSOCIATED WITH THE DEVELOPMENT OF ANY END USER APPLICATIONS.

USERS OF THE END USER APPLICATION ARE NOT ELIGIBLE FOR OBTAINING TECHNICAL SUPPORT FROM IMAGINE IT UNDER ANY CIRCUMSTANCES.

THIS SERVICE WILL BE PROVIDED ACCORDING TO IMAGINE IT'S TECHNICAL SUPPORT POLICIES FROM TIME TO TIME, AND IS LIMITED TO SUCH TIME AS IMAGINE IT SUPPORTS THE SOFTWARE OR OWNS THE SOFTWARE.

TECHNICAL SUPPORT MAY BE PROVIDED BY MEANS OF E-MAIL, FAX, BULLETIN BOARDS, ON-LINE SERVICES, WRITTEN CORRESPONDENCE OR TELEPHONE AT IMAGINE IT'S SOLE DISCRETION.

OWNERSHIP

IMAGINE IT SHALL REMAIN THE OWNERS OF ALL RIGHTS, TITLE AND INTEREST IN THE SOFTWARE

THIS LICENSE DOES NOT CONFER ANY OWNERSHIP RIGHTS TO THE LICENSEE

THE LICENSEE AGREES NOT TO USE IMAGINE IT'S NAME IN ANY WAY OR FORM

THE LICENSEE AGREES NOT TO IMPLY THAT ANY END USER APPLICATIONS IT DEVELOPS ARE EITHER APPROVED OR OTHERWISE SANCTIONED BY IMAGINE IT

THE LICENSEE AGREES TO CARRY A COPYRIGHT NOTICE IN THE ABOUT BOX OR HELP FILE OF EACH AND EVERY COPY OF ANY END USER APPLICATIONS IT DEVELOPS WITH THE HELP OF THIS SOFTWARE, WITH THE FOLLOWING WORDS :-

"PORTIONS OF THIS APPLICATION PROVIDED BY IMAGINE IT LIMITED, UNITED KINGDOM, COPYRIGHT (C) 1996-1997 IMAGINE IT LIMITED. ALL RIGHTS RESERVED"

TERMINATION

THIS AGREEMENT SHALL BE TERMINATED IN WRITING BY IMAGINE IT AT ANY TIME IF THE LICENSEE BREACHES ANY TERMS AND CONDITIONS CONTAINED IN THIS AGREEMENT OR CARRIES OUT ANY ACTIONS EXPRESSLY PROHIBITED IN THIS AGREEMENT, AND CONTINUES TO BE IN BREACH FOR 30 DAYS AFTER WRITTEN NOTICE IS GIVEN TO THE LICENSEE BY IMAGINE IT OR ITS REPRESENTATIVES.

UPON TERMINATION LICENSEE SHALL RETURN ALL COPIES OF THE SOFTWARE, ITS COMPONENTS AND DOCUMENTATION TO IMAGINE IT.

ALL RIGHTS GRANTED TO THE LICENSEE IN THIS AGREEMENT SHALL CEASE UPON TERMINATION OF THIS AGREEMENT EXCEPT FOR THOSE APPLYING TO ANY COPIES OF ANY END USER APPLICATIONS ALREADY PROPERLY DISTRIBUTED AND LICENSED WITHIN THE TERMS OF THIS AGREEMENT, PRIOR TO TERMINATION OF THIS AGREEMENT.

CONFIDENTIALITY

THE LICENSEE ACKNOWLEDGES THAT THE SOFTWARE AND ITS DOCUMENTATION AND DESIGN CONSTITUTE CONFIDENTIAL AND PROPRIETARY INFORMATION BELONGING TO IMAGINE IT.

THE LICENSEE AGREES NOT TO DISCLOSE ANY OF THIS INFORMATION TO OTHER THIRD PARTIES EXCEPT FOR ITS EMPLOYEES WHO ARE ENGAGED IN USING THE SOFTWARE AND ARE BOUND BY THESE SAME TERMS THROUGH A SEPARATE WRITTEN AGREEMENT WITH LICENSEE, OR EXCEPT WHERE REQUIRED TO DISCLOSE THE INFORMATION BY LAW.

WARRANTIES AND DISCLAIMERS

THE SOFTWARE IS PROVIDED "AS IS". IMAGINE IT DOES NOT WARRANT THAT THE SOFTWARE WILL OPERATE WITHOUT ERRORS, OR THAT IT WILL MEET ALL OF THE LICENSEE'S REQUIREMENTS.

ALL WARRANTIES, EXPRESSED OR IMPLIED, ARE EXCLUDED FROM THIS AGREEMENT INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE BUT NOT EXCLUDING ANY STATUTARY RIGHTS PROVIDED FOR BY LAW.

IMAGINE IT SHALL NOT BE LIABLE FOR ANY DAMAGES INCLUDING BUT NOT LIMITED TO INCIDENTAL OR CONSEQUENTIAL LOSS, LOSS OF PROFIT, LOSS OF SALES, INJURY, DEATH, LOSS OF OR DAMAGE TO PROPERTY OR ANY OTHER LOSS ARISING DIRECTLY OR INDIRECTLY FROM THE USE OF THE SOFTWARE.

IMAGINE IT'S SOLE REMEDY TO THE LICENSEE, AT IMAGINE IT'S SOLE DISCRETION SHALL BE LIMITED TO EITHER THE REPAIR OR REPLACEMENT OF FAULTY SOFTWARE OR THE REPAYMENT OF THE LICENSE FEE PAID BY THE LICENSEE UPON RETURN OF ALL COPIES OF THE SOFTWARE. IMAGINE IT'S TOTAL LIABILITY SHALL NOT EXCEED THE AMOUNTS PAID BY LICENSEE FOR THE PURCHASE OF THIS LICENSE.

COPYRIGHT (C) 1996-1997 IMAGINE IT LIMITED, UNITED KINGDOM, ALL RIGHTS RESERVED.

MICROSOFT IS A REGISTERED TRADEMARK OF MICROSOFT CORPORATION.

IMAGINE IT, THE IMAGINE IT LOGO AND iSound, ARE TRADEMARKS OF IMAGINE IT LIMITED

APPLICABLE LAW

THIS AGREEMENT SHALL BE GOVERNED BY TO THE LAWS OF ENGLAND.

END

In order to use **iSound** in your applications you need to distribute the OCX with your application. In addition you also need to ship some Microsoft® shared libraries which the OCX uses at run-time.

Before distributing any files which have been shipped with this product please read the license agreement and ensure that you comply with all its terms and conditions. Note that you do not have a license to distribute any Microsoft® files as part of this license and you should ensure that you have the appropriate licenses from Microsoft. You should not distribute these third party components from the disks supplied with this product.

Shipping iSound

To ship **iSound** to your customers you MUST ship the following files - you may NOT ship any other files belonging to this product under any circumstances.

iSound.OCX

iSound.RTL

You should install these files on the target system in the Windows System Directory, for example

<WindowsDirectory> \ System \

After installing these files and the other Microsoft® files identified below, you should register iSound on the target system,. By running the following command :-

```
REGSVR32.EXE /s iSound.OCX
```

Shipping Microsoft® Components

The following files are required on the target system :-

MFC42.DLL

OLEPRO32.DLL

These should be installed in the target system's Windows System Directory but only if these files are either not already installed or they are a later version than those already on the target system.

When you install and register a control, you should also register OLEPRO32.DLL. Using the following command :-

```
REGSVR32.EXE /s OLEPRO32.DLL
```

Perform this registration step only if you need to install OLEPRO32.DLL. If the DLL is installed already, you should assume that it has been registered.

You should also register MFC42.DLL. Unlike OLEPRO32.DLL, you should always register this DLL, even if it is already installed. To register this DLL run the following command :-

```
REGSVR32.EXE /s MFC42.DLL
```

To UnInstall **iSound** from your development system please follow these instructions :-

1. Un-Register the OCX by clicking the Un-register iSound OCX icon or running the following command

```
REGSVR32.EXE /u iSound.OCX
```

2. Run the Add / Remove programs applet from the Control Panel and select the **iSound** component to be removed.

iSound consists of a number of objects that provide access to the core DirectSound functions.   These objects are modelled largely around the DirectSound interfaces.

Each iSound object presents a number of properties and methods which largely correspond to the DirectSound interfaces and functions.

In addition we have also created some new objects that will make your programming task easier.

iSound

The iSound object provides the main interface to DirectSound.   You access all other objects via iSound methods and properties.   iSound is synonymous with the IDirectSound interface.

iSound Properties

iSound Methods

iSoundBuffer

The iSoundBuffer object is synonymous with the IDirectSoundBuffer interface.

iSoundBuffer Properties

iSoundBuffer Methods

i3DBuffer

An i3DBuffer object represents the 3D sound parameters of an iSoundBuffer.   It is similar to the DirectSound DS3DBUFFER structure.   iSound OCX does not implement any object analogous to the IDirectSound3DBuffer interface - instead you can directly manipulate 3D sound using the i3DBuffer properties.

i3Dbuffer Properties

i3DListener

An i3DListener object represents the 3D parameters of the sound listener.   It is similar to the DirectSound DS3DLISTENER structure.   iSound OCX does not implement any object analogous to the IDirectSound3DListener interface - instead you can directly manipulate the 3D listener using the i3DListener properties.

i3DListener Properties

iCaps

The iCaps object contains the capabilities of a sound device.   It is similar to the DirectSound DSCAPS structure.

iCaps Properties

iBCaps

The iBCaps object contains the capabilities of a sound buffer. It is similar to the DirectSound DSBCAPS structure.

iBCaps Properties

iDevice

The iDevice object is special to iSound. It provides information about sound devices and can be used to enumerate display devices on the target system.

iDevice Properties

iWaveFormatEx

The iWaveFormatEx object is provided by iSound OCX as a helper object. It is similar to the Win32 WAVEFORMATEX structure which is used in DirectSound to describe sound buffers.

iWaveFormatEx Properties

In order to help you get started quickly, we have provided sample programs that show how to use iSound OCX.

There are samples for Visual Basic as well as Visual C++

In the installed samples directory you will find the following :

Samples Directory

VB	- contains Visual Basic samples
MFC	- contains Visual C++ / MFC samples

You will also find in these directories the following definitions files that will be useful for developing your applications. These files contain constant definitions required by iSound OCX

iSoundConstants.h	C++ constants
iSoundConstants.bas	Visual Basic constants

iSound supports the following methods:

CloseDevice	Close all open devices
Compact	Re-organise buffer memory
CreateBufferDesc	Create an iBufferDesc object
CreateCaps	Create an iCaps object
CreateSoundBuffer	Create an iSoundBuffer object
CreateWfxFormat	Create an iWaveFormatEx object
DuplicateSoundBuffer	Duplicate a sound buffer
EnumDevices	Enumerate all sound devices
OpenDevice	Open a sound device for use
SetCooperativeLevel	Set the application co-operative level

Closes all open devices

Syntax

controlname.**CloseDevice**( )

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

Long

Parameters

None

Remarks

Closes all open devices.   Call this method as part of your application clean up.

Direct X Compatibility

This method has no Direct X equivalent

See Also

None

Adjusts sound memory, if any, into a contiguous block so that the largest portion of free memory is available.

Syntax

controlname.**Compact**( )

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

Long

Parameters

None

Remarks

Only available when using exclusive mode.

Returns DS_OK if successful, or one of the following error values otherwise:

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

DSERR_UNINITIALIZED

Direct X Compatibility

This method is equivalent to the IDirectSound::Compact ( ) method

See Also

None

Creates an iBufferDesc object

Syntax

controlname.**CreateBufferDesc**( )

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

iBufferDesc objBufferDesc

Parameters

None

Remarks

Use this method to create an iBufferDesc object.   The returned object can be used to then create sound buffers.

Direct X Compatibility

This method has no Direct X equivalent.

[illegible] Also

None

Creates an iCaps object

Syntax

controlname.**CreateCaps**( )

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

iCaps objCaps

Parameters

None

Remarks

Use this method to create an iCaps object. iCaps properties can be used to examine the capabilities of the sound hardware.

Direct X Compatibility

This method has no Direct X equivalent.

See Also

None

Creates an iSoundBuffer object.

Syntax

controlname.**CreateSoundBuffer** (Object oBufferDesc, String sWavFileName)

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

iSoundBuffer objSoundBuffer	An iSoundBuffer Object

Parameters

object oBufferDesc	An iBufferDesc object that contains the description of the sound buffer to be created. An iBufferDesc object can be created using the CreateBufferDesc method
string sWavFileName	Optional full path and filename of a wav file to load into the created buffer

Remarks

Use this method to create sound buffers.

The sWavFileName parameter is provided so that a wav file can be loaded into the newly created sound buffer. Set this parameter to NULL or "" (empty string) if you do not want to load a wav file. Otherwise set this parameter to the full pathname of the wav file you want to load.

Returns DS_OK if successful, or one of the following error values otherwise:

DSERR_ALLOCATED

DSERR_BADFORMAT

DSERR_INVALIDPARAM

DSERR_NOAGGREGATION

DSERR_OUTOFMEMORY

DSERR_UNINITIALIZED

DSERR_UNSUPPORTED

Direct X Compatibility

This method is equivalent to the IDirectSound::CreateSoundBuffer ( ) method

See Also

CreateBufferDesc method

Creates an iWaveFormatEx object.

Syntax

controlname.**CreateWfxFormat**( )

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

iWaveFormatEx objWaveFormatEx

Parameters

None

Remarks

Use this method to create an iWaveFormatEx object. An iWaveFormatEx object can be used wherever DirectSound requires a WAVFORMATEX structure.

Direct X Compatibility

This method has no Direct X equivalent, however an iWaveFormatEx object is similar to the Win32 WAVFORMATEX structure.

See Also

iWaveFormatEx Object

Creates a new iSoundBuffer object that uses the same buffer memory as the original object.

Syntax

controlname.**DuplicateSoundBuffer**(Object oSrcBuffer)

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

iSoundBuffer objSoundBuffer

Parameters

Object oSrcBuffer	An iSoundBuffer object to be duplicated.

Remarks

Returns a new iSoundBuffer Object that is a duplicate of the oSrcBuffer object. The new object can be used just like the original.

LastErrorCode is set to DS_OK if successful, or one of the following error values otherwise:

DSERR_ALLOCATED

DSERR_INVALIDCALL

DSERR_INVALIDPARAM

DSERR_OUTOFMEMORY

DSERR_UNINITIALIZED

Direct X Compatibility

This method is equivalent to the IDirectSound::DuplicateSoundBuffer ( ) method

See Also

None

Enumerates all sound devices

Syntax

controlname.**EnumDevices**( )

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

Short	numDevices	The number of devices found

Parameters

None

Remarks

This method enumerates all sound devices found on the system. The return value indicates the number of devices found. After calling this method you can use the DeviceList property to access any of the enumerated devices.

VB Example

```
n = iSound1.EnumDevices
If (n > 0) then   ‘ at least 1 device found
      ‘ open the first device
      iSound1.OpenDevice ( iSound1.DeviceList ( 0 ) )
End If
```

Direct X Compatibility

This method has no Direct X equivalent

See Also

DeviceList property

Opens a sound device

Syntax

controlname.**OpenDevice** (Object objDevice)

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

Long

Parameters

Object objDevice	An iDevice Object

Remarks

Use this method to open a sound device. Use the EnumDevices method to first enumerate all devices. You can then use the DeviceList property to open any one of the enumerated devices. Alternatively you can pass a NULL object in the objDevice parameter to open the default device.

VB Example

```
n = iSound1.EnumDevices

If (n > 0) then    ‘ at least 1 device found

      ‘ open the first device

      iSound1.OpenDevice ( iSound1.DeviceList ( 0 ) )

End If
```

Direct X Compatibility

This method has no Direct X equivalent. However it is similar to using the DirectSoundCreate ( ) method

See Also

EnumDevices method

DeviceList property

Sets the cooperative level of the application for this sound device.

Syntax

controlname.**SetCooperativeLevel**(OLE_HANDLE hWnd, Long lFlags)

controlname is the name of the **iSound** Object, for example, iSound1.

Return Value

Long

Parameters

hWnd	Window handle of the application.
lFlags	Flags indicating the co-operative level to set

Remarks

Returns DS_OK if successful, or one of the following error values otherwise:

DSERR_ALLOCATED
DSERR_INVALIDPARAM
DSERR_UNINITIALIZED
DSERR_UNSUPPORTED

The cooperative level can be one of:

DSSCL_EXCLUSIVE
DSSCL_NORMAL
DSSCL_PRIORITY
DSSCL_WRITEPRIMARY

Direct X Compatibility

This method is equivalent to the IDirectSound::SetCooperativeLevel ( ) method

See Also

None

iSound supports the following properties:

Caps	Capabilities of a sound device
DeviceList	List of enumerated sound devices
LastErrorCode	Last error as a code
LastErrorString	Last error as a message
SpeakerConfig	Speaker configuration

Gets the capabilities of a sound device.

Syntax

controlname**.Caps** [= *Caps*]

controlname is the name of the **iSound** Object, for example, iSound1.

Type

iCaps objCaps	an iCaps Object

Remarks

This property represents an object of type iCaps. An iCaps object contains properties that describe the capabilities of the hardware device.

If there is an error objCaps is set to NULL and LastErrorCode is set to one of the following error values:

DSERR_GENERIC

DSERR_INVALIDPARAM

DSERR_UNINITIALIZED

Visual Basic Example

```
Dim objCaps as Object
iSOund1.OpenDevice( )
Set objCaps = iSound1.Caps
if (Not IsNull objCaps) then
      ‘ success, we can now check properties of objCaps
else
      ‘ error, check LastErrorCode and LastErrorString properties
endif
```

Direct X Compatibility

This property is the equivalent of IDirectSound::GetCaps

See Also

iCaps Object

List of enumerated sound devices

Syntax

controlname**.DeviceList** [n]

controlname is the name of the **iSound** Object, for example, iSound1.

Type

iDevice objDevice

Remarks

This property array represents a list of sound devices found on the system.   Initialise the DeviceList first by calling the EnumDevices method.

Visual Basic Example

```
n = iSound1.EnumDevices ( )
if ( n > 0 ) then
      ‘ open the first device
      iSound1.OpenDevice ( iSound1.DeviceList ( 0 ) )
endif
```

Direct X Compatibility

This property has no Direct X equivalent.

See Also

EnumDevices method

Represents the latest error that occurred.

Syntax

controlname**.LastErrorCode**

controlname is the name of the **iSound** Object, for example, iSound1.

Type

Long

Remarks

Use this property to check for any errors after using any iSound methods and properties. LastErrorCode is always set to DS_OK for no error. To get a verbose error message use the LastErrorString property.

Direct X Compatibility

There is no Direct X equivalent to this property

See Also

LastErrorString property

An error message representing the latest error that occurred.

Syntax

controlname**.LastErrorString**

controlname is the name of the **iSound** Object, for example, iSound1.

Type

String

Remarks

Use this property to get a verbose error message representing the last error.

Direct X Compatibility

There is no Direct X equivalent to this property

See Also

LastErrorCode property

Get or set the speaker configuration

Syntax

controlname**.SpeakerConfig** [= *SpeakerConfig*]

controlname is the name of the **iSound** Object, for example, iSound1.

Type

Long

Remarks

Use this method to get or set the configuration of the system’s speakers.   Allowed values are:

DSSPEAKER_HEADPHONE

DSSPEAKER_MONO

DSSPEAKER_QUAD

DSSPEAKER_STEREO

DSSPEAKER_SURROUND

Direct X Compatibility

This property is the equivalent to the IDirectSound::GetSpeakerConfig ( ) and IDirectSound::SetSpeakerConfig ( ) methods

See Also

None

iSoundBuffer supports the following methods:

CommitSettings	Commits 3D sound settings
Play	Begins playing a sound buffer
Restore	Restores memory for this sound buffer
Stop	Stops playing the sound buffer
Write	Writes data directly into the sound buffer

Commits any 3D sound settings made since the last call to this method.

Syntax

iSoundBuffer.**CommitSettings**( )

Return Value

Long

Parameters

None

Remarks

Always call this method after making any changes to the i3Dbuffer and i3Dlistener objects associated with this sound buffer.   Unlike DirectSound, there is no way to modify 3D settings immediately using iSound OCX – you must call this method to have your changes take effect.

Returns DS_OK if successful, or DSERR_INVALIDPARAM otherwise.

Direct X Compatibility

This property is equivalent to the IDirectSound3DListener::CommitDeferredSettings ( )

See Also

i3Dlistener object

i3Dbuffer object

Plays the sound buffer from the current position.

Syntax

iSoundBuffer.**Play**(Long lFlags)

Return Value

Long

Parameters

lFlags Flags specifying how to play the buffer. This should be set to either 0 or DSBPLAY_LOOPING.

Remarks

Returns DS_OK if successful, or one of the following error values otherwise:

DSERR_BUFFERLOST

DSERR_INVALIDCALL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::Play ( ) method

See Also

None

Restores the memory allocation for a lost sound buffer.

Syntax

iSoundBuffer.**Restore**( )

Return Value

Long

Parameters

None

Remarks

Returns DS_OK if successful, or one of the following error values otherwise:

DSERR_BUFFERLOST

DSERR_INVALIDCALL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::Restore ( ) method

See Also

None

Stops playing the sound buffer.

Syntax

iSoundBuffer.**Stop**( )

Return Value

Long

Parameters

None

Remarks

Returns DS_OK if successful, or one of the following error values otherwise:

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::Stop ( ) method

See Also

None

Writes data directly ino the sound buffer.

Syntax

iSoundBuffer.**Write** (variant vBuffer, long lSize)

Return Value

Long

Parameters

vBuffer	A buffer containing the data to be written
lSize	size in bytes of the data in vBuffer

Remarks

This method allows you to write directly into the sound buffer. It encapsulates the DirectSound Lock () and UnLock () methods.

The data is written from the current write cursor position. If the data overflows past the end of the buffer the remaining data is written to the beginning of the sound buffer.

You can write data while the buffer is playing.

This method fails if lSize is greater than the size of the sound buffer.

Direct X Compatibility

This method has no Direct X equivalent

See Also

None

iSoundBuffer supports the following properties:

Caps	Sound Buffer capabilities
CurrentPlayCursorPosition	The current position of the play cursor
CurrentWriteCursorPosition	The current position of the write cursor
Format	Sound format
Frequency	Sound frequency
Pan	Sound pan (or balance)
Sound3DBuffer	The associated 3D sound buffer
Sound3DListener	The associated 3D sound listener (only applies to the primary sound buffer)
Volume	Sound volume

The capabilities of the iSoundBuffer object.

Syntax

iSoundBuffer.**Caps**

Type

iBCaps objCaps	an iBCaps object

Remarks

This property represents capabilities of a SoundBuffer. The capabilities are returned as an iBCaps object. You can access iBCaps properties to examine individual capabilities.

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::GetCaps ( ) method and the DSBCAPS structure

See Also

iBCaps object

The location in this buffer where a sound is being played.

Syntax

iSoundBuffer.**CurrentPlayCursorPosition**

Type

Long

Remarks

Returns the current play positon within the sound buffer.

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::GetCurrentPosition ( ) method

See Also

None

The location in this buffer where it is safe to change data.

Syntax

iSoundBuffer.**CurrentWriteCursorPosition**

Type

Long

Remarks

Returns the position in the sound buffer where you can safely modify the sound data.

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::GetCurrentPosition ( ) method

See Also

None

Gets or sets the format of the sound data in the buffer.

Syntax

iSoundBuffer.**Format** [= *Format*]

Type

iWaveFormatEx Object	an iWaveFormatEx object

Remarks

This property represents the format of the sound buffer as an iWaveFormatEx object.

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::GetFormat ( ) method

See Also

iWaveFormatEx object

Get or set the frequency of the sound buffer

Syntax

iSoundBuffer.**Frequency** [= *Frequency*]

Type

Long

Remarks

Use this property to change the frequency at which the sound buffer is played. The frequency is measured in Hertz and must be between 100 and 100,000. You cannot change the frequency of the primary buffer

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::GetFrequency ( ) and IDirectSoundBuffer::SetFrequency ( ) methods

See Also

None

Get or set the relative volume between the right and left channels (balance)

Syntax

iSoundBuffer.**Pan** [= *Pan*]

Type

Long

Remarks

Use this property to change the relative volume of the right and left channels. The pan value is measured in 100 th of a decibel. A value of 0 indicates both left and right channels are at the full volume. A value of 100 indicates that the left channel is attenuated by 1.0 decibels and the right channel is at full volume. Conversely a value of –100 indicates that the right channel is attenyuated by 1.0 decibels and the left channel is at full volume. The maximum value you can set is + / - 10,000

You cannot change the pan value for the primary buffer

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::GetPan IDirectSoundBuffer::SetPan methods

See Also

None

Gets the 3D sound buffer associated with this sound buffer

Syntax

iSoundBuffer.**Sound3DBuffer** [= *Sound3DBuffer*]

Type

i3DBuffer Object	An i3DBuffer Object

Remarks

The i3DBuffer object provides access to the 3D parameters of a sound buffer if supported. You can set the properties of the i3DBuffer object to create 3D sound effects.

Only sound buffers with 3D capabilities have a 3DSound object. For all other types of sound buffers this property is NULL.

Direct X Compatibility

This property is equivalent to the DS3DBUFFER structure

See Also

i3DBuffer Object

Gets the 3D sound listener associated with the primary sound buffer

Syntax

iSoundBuffer.**Sound3DListener** [= *Sound3DListener*]

Type

i3DListener Object	An i3DListener Object

Remarks

A 3D listener represents the person who hears sounds generated by sound buffer objects in 3D space. The i3DListener object controls the listener's position and velocity in 3D space.

You can set the properties of the i3DListener object to modify the listeners 3D parameters.

Only the primary sound buffer with 3D capabilities has a 3DListener object. For all other types of sound buffers this property is NULL.

Direct X Compatibility

This property is equivalent to the DS3DLISTENER structure

See Also

i3Dlistener Object

Gets or sets the volume of this sound buffer

Syntax

iSoundBuffer.**Volume** [= *Volume*]

Type

Long

Remarks

Use this property to change the volume at which the sound buffer is played. The volume is measured in 100 th of a decibel and must be between 0 and –10,000. DirectSound does not currently support amplification, so the volume can only be attenuated. 0 indicates no change in volume and –10,000 indicates the volume is attenuated by 100 decibels

Direct X Compatibility

This property is equivalent to the IDirectSoundBuffer::GetVolume ( ) and IDirectSoundBuffer::SetVolume ( ) methods

See Also

None

iWaveFormatEx supports the following properties:

AvgBytesPerSec	Average data transfer rate
BitsPerSample	Bits per sample
BlockAlign	Block alignment
Channels	Number of channels
FormatTag	Format Tag
SamplesPerSec	Samples per second
Size	Size of extra information

Average data-transfer rate, in bytes per second

Syntax

iWaveFormatEx.**AvgBytesPerSec** [= *AvgBytesPerSec*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the AvgBytesPerSec member of the WIN32 WAVEFORMATEX Structure

See Also

None

Bits per sample for the wFormatTag format type

Syntax

iWaveFormatEx.**BitsPerSample** [= *BitsPerSample*]

Type

Short

Remarks

None

Direct X Compatibility

This property is equivalent to the BitsPerSample member of the WIN32 WAVEFORMATEX Structure

See Also

None

Block alignment, in bytes

Syntax

iWaveFormatEx.**BlockAlign** [= *BlockAlign*]

Type

Short

Remarks

None

Direct X Compatibility

This property is equivalent to the BlockAlign member of the WIN32 WAVEFORMATEX Structure

See Also

None

Number of channels in the waveform-audio data

Syntax

iWaveFormatEx.**Channels** [= *Channels*]

Type

Short

Remarks

None

Direct X Compatibility

This property is equivalent to the Channels member of the WIN32 WAVEFORMATEX Structure

See Also

None

Waveform-audio format type

Syntax

iWaveFormatEx.**FormatTag** [= *FormatTag*]

Type

Short

Remarks

None

Direct X Compatibility

This property is equivalent to the FormatTag member of the WIN32 WAVEFORMATEX Structure

See Also

None

Sample rate, in samples per second (hertz), that each channel should be played or recorded

Syntax

iWaveFormatEx.**SamplesPerSec** [= *SamplesPerSec*]

Type

Long

Remarks

None

Direct X Compatibility

This property is equivalent to the SamplesPerSec member of the WIN32 WAVEFORMATEX Structure

See Also

None

Size, in bytes, of extra format information appended to the end of the WAVEFORMATEX structure

Syntax

iWaveFormatEx.**Size** [= *Size*]

Type

Short

Remarks

In the current version of iSound OCX this must be set to 0.

Direct X Compatibility

This property is equivalent to the Size member of the Win32 WAVEFORMATEX Structure

See Also

None