Microsoft® DirectX™ 3
ƒƒƒƒƒƒŠ”ƒƒƒ

,,ƒƒƒƒƒƒ,‹ ,,,,, •, — ,,,• ,,,,,,,, ,, •’,‰Ž– –
,,,ƒ ƒ,“,‹ ,,,,,,Œ,‰‹,,,,,, ,,ƒƒƒƒƒƒ,‘•,,,ˆ•, Microsoft Corporation, –,,,‹‰
,,,, “‹“ ‹Š“,,,,,,•Ž •–,,,• “’,,,,,,,,,
Microsoft , ,,ƒƒƒƒƒƒ,‹ ,,,,, ƒƒƒƒ ƒƒƒ,“‹,,,, ’,“‹ • ’ Œ ,,‘,’“ —
Œ,•Ž,,,, ,,ƒƒƒƒƒƒ, Microsoft , –,,,‹‰,,,, “‹ • ’ Œ ,,‘,’“ —Œ,‹‰,—
,,,,,,,,

Microsoft ActiveMovie Direct3D DirectDraw DirectInput DirectPlay DirectSound DirectX MS-DOS Win32 Windows ,,,Windows NT, • Microsoft Corporation,• ,,,,,‘, ,,,, •,,,

,,‘, •–,‰Ž–,ŠŽ, •,,,

3

DirectSound

•’...

DirectSound,,,,

The Microsoft® DirectSound®
(API) DirectX™ 3 Software Development Kit (SDK)のDirectSound
のの

DirectX 3
DirectX の
DirectSound
の

DirectSound*ƒ□ƒƒƒƒƒ*

のDirectSound のの DirectXのの

- *ƒ ƒƒƒƒƒŠ—*
- *ƒƒƒƒƒƒ ƒƒƒ*
- *ƒƒƒƒƒƒ□ƒƒƒƒ□ƒƒƒ*
- *ƒƒƒƒ□ƒƒƒƒ*
- *‹’ƒƒƒ*
- *ƒƒƒƒ‚“□*

*ƒ□ƒƒƒƒƒ*Š—

ののMicrosoft のの MIDI DirectSound のの MIDI
Windows 95 API の MIDI MIDI
ののMIDI Microsoft Win32® Software
Development Kit (SDK)

DirectSound
のの のDirectX 3 SDK の
DirectSound
の

DirectSound *IDirectSound* *IDirectSoundBuffer* COM ののDirectX 3
SDK の COM のの

DirectSound の DirectSound の **IDirectSound::CreateSoundBuffer**
DirectSoundBuffer DirectSoundBuffer の
DirectSoundBuffer の DirectSoundBuffer
DirectSound

DirectSound RAM
(DMA) (I/O のの
)ののの

のの1 の

の

ƒƒƒƒƒƒ□ƒƒƒ

の DirectSound DirectSound *I DirectSound* (COM)ののの

のDirectSound ののの

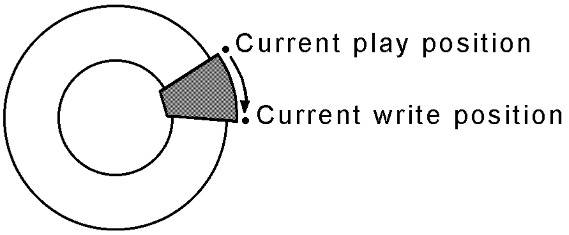
DirectSound *IDirectSoundBuffer*

ののDirectSound ののの DirectSound のののの

ののののDirectSound のののの

の

DirectSound



DirectSound

のDirectSound DirectSoundBuffer

- *DirectSoundƒƒƒƒƒƒ*
- *DirectSoundBuffer ƒƒƒƒƒƒ*

DirectSound*ƒƒƒƒƒƒ*

の DirectSound *IDirectSound* DirectSound **DirectSoundCreate** **IDirectSound** **DirectSoundEnumerate**

Windows
の
DirectSound

Note DirectSound,*ƒƒƒƒƒƒƒ*,**IDirectSound**,*ƒƒƒƒ*,*IDirectSoundBuffer*ƒƒƒ ƒƒƒƒ ,’‹,,, CŒŒ,ƒƒƒ’‹,Š,,,,

DirectSoundBuffer*ƒƒƒƒƒƒ*

DirectSoundBuffer *IDirectSoundBuffer* **IDirectSound::CreateSoundBuffer** DirectSoundBuffer **IDirectSoundBuffer**

のの DirectSound

IDirectSoundBuffer::Lock

IDirectSoundBuffer::Play のの DirectSound

IDirectSoundBuffer::Play の

のの
の

DirectSoundBuffer **IDirectSound::CreateSoundBuffer** DSBCAPS_STATIC (の)

PCI のの
ののの
DirectSound

ƒƒƒƒƒƒ□*ƒƒƒƒ*□*ƒƒƒ*

DirectSound
IDirectSound::GetCaps
DirectSound

ƒƒƒƒ□*ƒƒƒƒ*

DirectSound
DirectSound HAL Windows の
HAL の
Windows のDirectSound HAL

- ƒ ƒƒƒ‹Š,‘,,ƒƒƒƒ ƒ,Ž“,‰•

- ƒ ƒƒƒ‹Š‚ ”•‰
- ƒ ƒƒƒƒ‚—Œ‚ ‚“’‚‘ ‚Ž
- ƒ ƒƒƒƒ‚–Œ‚ ‚‘ —‹‚‹”

DirectSound の DirectSound の Windows 95 Windows NT® Windows 3.1 ののDirectSound のの

‹’ƒƒƒ

DirectSound 4 **IDirectSound::SetCooperativeLevel**

の **IDirectSoundBuffer::SetFormat** **IDirectSound::Compact** のの

DirectSound **IDirectSoundBuffer::SetFormat** **IDirectSound::Compact**

DirectSound DirectSound wave wave **IDirectSoundBuffer::SetFormat** ののの (DirectSound wave)

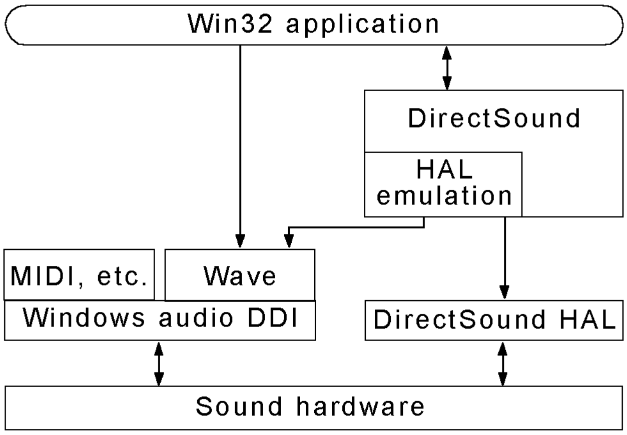
の DirectSound **IDirectSoundBuffer::Lock**

(の**IDirectSoundBuffer::Restore**) の の

IDirectSoundBuffer::GetFormat, **IDirectSoundBuffer::SetFormat**, **IDirectSoundBuffer::GetVolume** のの **IDirectSoundBuffer::Lock** **IDirectSoundBuffer::Play** の

ƒƒƒƒ‚“□

の DirectSound ののの



DirectSound HAL
HAL のの
HAL の

DirectSound の DirectSound
DirectSound
DirectSound の HAL の

DirectSound

のの
DirectSoundCreate *IDirectSound*
IDirectSoundBuffer
IDirectSound::Release
の Win32 SDK の

のの
PlaySound **WaveOut**

Note Microsoft Video for Windows, Œ avi(audio visual interleaved)*ƒƒƒƒ*,*ƒ ƒƒƒ ƒƒƒƒ*, —,,,,*ƒƒ ƒƒƒ ƒ ƒ ƒƒƒ*Š ,Ž—
,,,, ,,,, *ƒƒƒƒ ƒƒƒ*,DirectSound,Ž—
,,,,avi*ƒƒƒƒ*, ,,,,,,, ,,*ƒ ƒƒƒ ƒƒƒƒ*, ,,,, “—
, avi*ƒƒƒƒ*, ,,,,DirectSound*ƒƒƒƒƒƒ*, ,,,,,,, Ž,Š ,*ƒƒ* ,•,

Œ ,,,,,, *ƒƒƒƒ ƒƒƒ*,DirectSound*ƒƒƒƒƒƒ*,‰•,,,,,, avi*ƒƒƒƒ*, ‘,**IDirectSound::Release**,Œ, ,, , *ƒƒƒƒ ƒƒƒ*,*ƒƒƒ*, —
Œ,DirectSound*ƒƒƒƒƒƒ*,,,DirectSoundBuffer*ƒƒƒƒƒƒ*, , ’, ,,

DirectSoundŠ—

の DirectSound の

- *DirectSound,“’*
- *3Dƒƒƒƒ*

DirectSound,“’

の DirectSound ののの

- *ƒƒƒƒƒ*
- *ƒ ƒƒƒƒ ƒƒƒƒƒ ƒƒƒ*
- *ƒƒƒƒƒ ƒƒƒƒ„ „„ƒƒƒƒ*

ƒƒƒƒƒ

DirectSound の
DirectSound のののDirectSound
の

20 のののののDirectSound の 20 ののののの DirectSound HEL (の DirectSound)

ののののののの DirectSound

ƒ□ƒƒƒƒ□ƒƒƒƒƒ□ƒƒƒ

DirectSound の

のの DirectSound

のの

のののの

ƒƒƒƒƒ□ƒƒƒƒ„□„„ƒƒƒƒ

DirectSound □□□□□□□□□□ □□□

□□□□□□□□□□ □□□

□□□

□□□

DirectSound の
(DirectSound

の □□□□□□□□□□

の□□□ ののののののののの

3Dƒƒƒƒ

DirectSound

IDirectSound3DBuffer *IDirectSound3DListener*

の

のの

3D のの

- *‰‘’ˆ‚”Ž*
- *ƒƒƒ*
- *ƒƒƒƒ ƒ ƒ*
- *’‹—‚ ’‹—*
- *ˆ’‚‘“*
- *Direct3D‚‚“*
- *‘’’ˆ‚‹——‘*
- *ƒƒƒ‰Œ‚ƒƒƒ‰Œ*

‰‘’ˆ‚”Ž

のののの

- *ƒƒƒ ƒ ‰‚ ‚•‘‚ƒƒƒ ‚‚—‚‚‚ ‚‚‰‚ ‚‚‚‚‚‚ ‚‚Œ ‚ƒ ƒƒƒ‚Œ‚‚‚‚*
- *“’‚‚‚ ƒƒƒ ‚‰•Œ‚‚”‚‚‚‰‚‚ƒƒƒ ‚‰Ž‚“’‚‚‚ Ž‚‚‚‚‚ ‚ ‚‚‚ (‚‚‚‚Ž‘ŽŠ‚ –1ƒƒ•‚‚‚)*
- *‰ ƒƒƒ ‚Œ‚‚‚‚‰‚ ‘‚‚‚‰‚”‚‚‚‚‚ ‰ ‚‚‚‚ ‚‚‚‚ ‚Ž‚•ŒŠ”‚ŠŽ‚‚ ‚‚‚ ƒƒƒ ‚‰‚‚‰‚‚ ‚ Ž‚“‚‰‚“‚‚‚‚ ‰ ‚‚‚*

のの DirectSound のの 3D

のののの

ƒƒƒ□

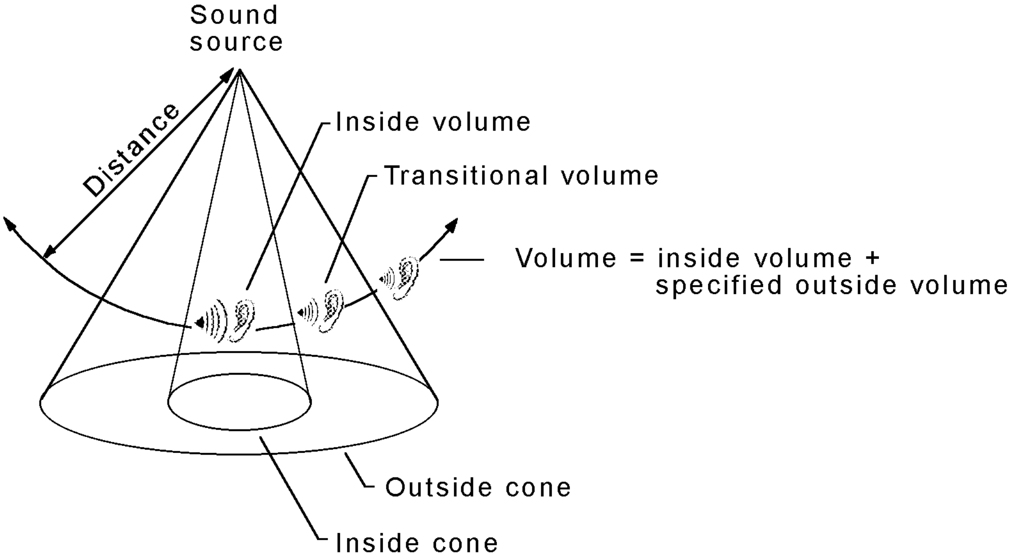
の 90 90 のののののDirectSound *IDirectSound3Dlistene*

ƒƒƒƒ□ƒ□ƒ

□□□□□□□□□

DirectSound の□□□□□□□□□ のの(DirectSound 0 ののの の)のの-10,000 のの

□

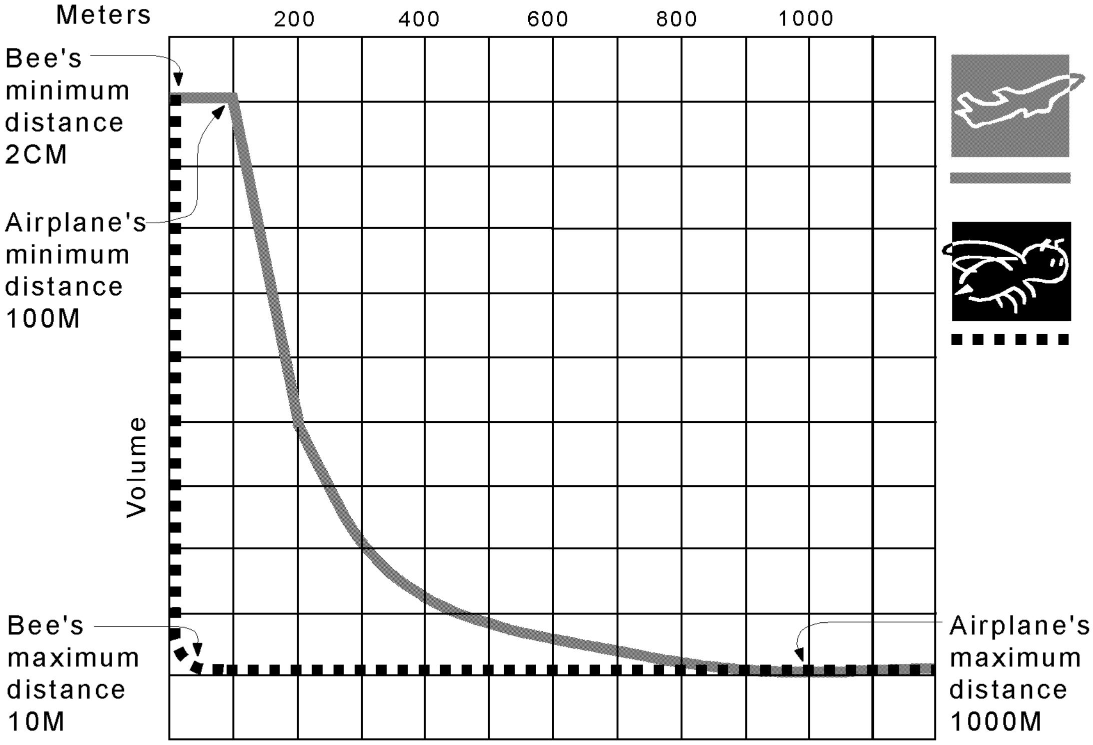


IDirectSound3Dbuffer
□□□□□□□□□ □□□□□□□□□ののの 0
のの

□’‹—,□’‹—

(0)ののののの

(16 のののの) 100 の
2 のの
200 4
ののの



^, ‘‘

の 3D 3D ののの3D 3D のの 3D
の
の
DirectSound の

ののののの
のの 3D
の

の
のの

Direct3D,,“□

IDirectSound3DBuffer *IDirectSound3Dlistene*
Direct3D™ Direct3D のの
Direct3D の **D3DVECTOR** **D3DVALUE**
IDirectSound3Dbuffer **IDirectSound3Dlistener**
Direct3D の DirectSound
(*Direct3D* *のの* *3D*)

3D の Direct3D の
D3DRMFRAMEMOVECALLBACK

‘’’^‚‹——‘

3D　　　　　　　ののの

DirectSound の 3D のののの 0.30480006096　1

ƒƒƒƒ‰Œ‚ƒƒƒƒ‰Œ

3D　　　　のの

DirectSound の 3D の

DirectSoundƒƒƒ□ƒƒƒƒŠ—

のの DirectSound

- *IDirectSoundƒƒƒ　ƒƒƒƒ*
- *IDirectSound3DBuffer ƒƒƒ　ƒƒƒƒ*
- *IDirectSound3DListener ƒƒƒ　ƒƒƒƒ*
- *IDirectSoundBuffer ƒƒƒ　ƒƒƒƒ*

IDirectSoundƒƒƒ□ƒƒƒƒ

DirectSound　DirectSoundBuffer
DirectSound　　　　　　　　　　*IDirectSoundBuffer*
　　　　IDirectSound のの

ののののののの

- *ƒƒƒƒ‚　”*
- *ƒƒƒƒ‚*
- *ƒƒ　ƒ‚Š‹　’*
- *ƒ　ƒƒƒƒƒƒƒŠ—*

ƒƒƒƒ‚□”

DirectSoundCreate　　　　　　DirectSound
IDirectSound::GetCaps

ƒƒƒƒ‚□□

DirectSoundCreate　　　　　　DirectSound
IDirectSound::CreateSoundBuffer
IDirectSound::DuplicateSoundBuffer

IDirectSound::SetCooperativeLevel
のの
DSSCL_NORMAL のの

ƒƒ□ƒ‚Š‹□’

IDirectSound
の **IDirectSound::GetSpeakerConfig**
IDirectSound::SetSpeakerConfig

ƒ□ƒƒƒƒƒƒŠ—

IDirectSound::Compact
の

IDirectSound3DBufferƒƒƒ□ƒƒƒƒ

IDirectSound3DBuffer の 3D のの

の **IDirectSound3DBuffer** ののの

- *IDirectSound3DBufferƒƒƒ ƒƒƒƒƒƒƒƒ‚Ž“*
- *ƒƒƒƒƒƒ ƒ‘*
- ’‹—’‚ ’‹—’
- ‘ *ƒ ƒ*
- ˆ’‚‘“
- *ƒƒƒƒ ƒƒƒƒƒƒƒƒ ƒ ƒ*

IDirectSound3DBufferƒƒƒ□ƒƒƒƒƒƒƒƒ‚Ž“

IDirectSound3DBuffer の 3D
IDirectSound::CreateSoundBuffer **DSBUFFERDESC**
dwFlags の DSBCAPS_CTRL3D
IDirectSoundBuffer::QueryInterface
IDirectSound3Dlistener の

```
// DSBCAPS_CTRL3D,Ž—,,lpDsbSecondary,□□,,
hr = lpDsbSecondary->QueryInterface(IID_IDirectSound3DBuffer,
        &lpDs3dBuffer);
    if(SUCCEEDED(hr)) {
    // ,,ƒƒƒƒ,3Dƒƒƒ□ƒ,□’,,
        .
        .
        .
}
```

DirectSound 3D の
3D の
3D

Note ƒƒ Œ,3D —,‹ ,, ,,,,
DSBCAPS_CTRL3D,DSBCAPS_CTRLPAN,—
•,Ž’,,,,,,DirectSound,—‹, ,Ž”,,

ƒƒƒƒƒƒ□ƒ‘□

3D の

の

IDirectSound3DBuffer

のの

IDirectSound3DBuffer::GetAllParameters

□’‹—’,□’‹—’

3D

のの

の **IDirectSound3DBuffer::SetMinDistance**
IDirectSound3DBuffer::GetMinDistance
の **IDirectSound3DBuffer::SetMaxDistance**
IDirectSound3DBuffer::GetMaxDistance

‘□ƒ□ƒ

( )ののの

3D の **IDirectSound3DBuffer::SetMode**
の *dwMode* の

ˆ’‚“

3D 3D の
DirectSound3DBuffer::SetPosition
IDirectSound3DBuffer::GetPosition

DirectSound
IDirectSound3DBuffer::SetVelocity
IDirectSound3DBuffer::GetVelocity
ののの

ƒƒƒƒ□ƒƒƒƒƒƒƒƒ□ƒ□ƒ

3D
IDirectSound3DBuffer
のの

IDirectSound3DBuffer

ののの

•ƒ ƒ‚Š“‚•Š

•“• Š•ƒ ƒ ƒƒƒ ƒ

ƒ□ƒ‚Š“‚•Š

の

IDirectSound3DBuffer::SetConeAngles
IDirectSound3DBuffer::GetConeAngles
IDirectSound3DBuffer::SetConeOrientation
IDirectSound3DBuffer::GetConeOrientation

の 360
の

“•□Š•ƒ□ƒ□ƒƒƒ□ƒ

の ののののの 100dB
の 0

の **IDirectSound3DBuffer::SetConeOutsideVolume**
IDirectSound3DBuffer::GetConeOutsideVolume

の
(の **IDirectSoundBuffer::GetVolume**)

IDirectSound3DListenerƒƒƒ□ƒƒƒƒ

3D 3D の
の *IDirectSound3DListener* の 3D

の **IDirectSound3Dlistener** のの
の

- *IDirectSound3Dlistenerƒƒƒ ƒƒƒƒ ƒƒƒƒ‚Ž“*
- *ƒƒƒƒƒƒ ƒ‘*
- *‚’‚ ’*
- *‹——‘*
- *ƒƒƒƒ —‘*
- *ƒƒƒ ‚ˆ’‚‘“*
- *ƒƒƒ ‚•Œ*
- *ƒ ƒƒƒ—‘*

IDirectSound3Dlistenerƒƒƒ□ƒƒƒƒ□ƒƒƒƒ‚Ž“

IDirectSound3DListener
3D
IDirectSound::CreateSoundBuffer **DSBUFFERDESC**

の**dwFlags** DSBCAPS_CTRL3D
IDirectSoundBuffer::QueryInterface
IDirectSound3Dlistener のの

```
// DSBCAPS_CTRL3D‚Ž—‚‚lpDsbPrimary‚□□‚‚

hr = lpDsbPrimary->QueryInterface(IID_IDirectSound3DListener,
    &lpDs3dListener);

if(SUCCEEDED(hr)) {
    // 3D‘□‚Ž□‚‚
    .
    .
    .
}
```

ƒƒƒ□ƒƒƒ□ƒ‘□

の 3D のの
IDirectSound3DListener
のの
IDirectSound3DListener::GetAllParameters
IDirectSound3DListener::SetAllParameters

□‚’‚□’

3D 3D の
のの *dwApply* DS3D_DEFERRED の
IDirectSound3DListener::CommitDeferredSettings ¡

Note ‚‚ ‚’‚ ’‚ **IDirectSound3DListener::CommitDeferredSettings**‚Œ‚ ‚‚‚ ‚ “‚ ’‚DS3D_IMMEDIATEƒƒƒ‚Ž—‚‚Œ‚ ‚‚ ‚‚ ‚‚‚ ‚’‚ƒƒƒ‚Ž— ‚‚ƒƒƒ ‘“‚(1,2,3)‚ ’‚ ‚‚‚‚“‚ƒƒƒ‚Ž—‚‚ƒƒƒ ‘“‚(4,5,6)‚ ’‚‚‚‚‚‚ ƒƒƒ ‘“‚(4,5,6) ‚‚‚ ‚‚Œ
DirectSound3DListener::CommitDeferredSettingsƒƒƒƒ‚Œ‚ ‚‚ ‘“‚(4,5,6) ‚‚‚

‹———‘

DirectSound の
の

IDirectSound3DListener::SetDistanceFactor *flDistanceFactor*
0.30480006096 (の1)のの

の **IDirectSound3DListener::GetDistanceFactor**
DS3D_DEFAULTDISTANCEFACTOR (1.0)の 1
(3.0,7.2,-20.9)のの 3m

7.2m 20.9m 2.0
6m 14.4m 41.8m

ƒƒƒƒ□—‘

DirectSound 3D
DirectSound
10 の
IDirectSound3DListener::SetDopplerFactor
0 10 の0 の
の1 2
2 3D
IDirectSound3DListener::GetDopplerFactor

ƒƒƒ□‚ˆ’‚“

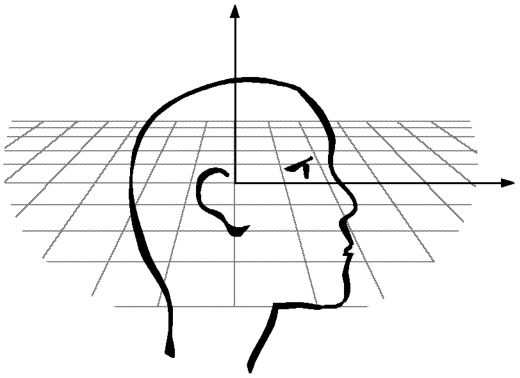
IDirectSound3DListener::SetPosition
IDirectSound3DListener::GetPosition 3D の

の **IDirectSound3DListener::SetVelocity**
IDirectSound3DListener::GetVelocity
のの

ƒƒƒ□‚•Œ

の 3D ののの DirectSound の

ののののののののののののののの



の **IDirectSound3DListener::SetOrientation**
IDirectSound3DListener::GetOrientation
(0,0,1.0) (0,1.0,0)

ƒ□ƒƒƒ—‘

DirectSound 10のの
IDirectSound3DListener::SetRolloffFactor
0 10の0
の 1

IDirectSoundBufferƒƒƒ□ƒƒƒƒ

IDirectSoundBuffer

Note DirectSoundBufferƒƒƒƒƒƒ‚‚‚‚ ‚‚DirectSoundƒƒƒƒƒƒ‚ — ‚‚‚ DirectSoundƒƒƒƒƒƒ‚‰•‚‚‚‚ ‚‚ƒƒƒƒƒƒ‚ ‚‚ƒƒƒƒ‚‚‚‰•‚‚ Ž ‚‚‚‚‚‚‚‚

のの のQの

- *Š—*
- *ƒƒƒƒŠ‹Š—*
- *•‚Ž“*
- *ƒƒƒŠ—*

□□Š—

の **IDirectSoundBuffer::Play**
IDirectSoundBuffer::Stop
IDirectSoundBuffer::Play
IDirectSoundBuffer::Stop

IDirectSoundBuffer::Lock □□□
□□□
IDirectSoundBuffer::Unlock

の
IDirectSoundBuffer::GetCurrentPosition
IDirectSoundBuffer::SetCurrentPosition

ƒƒƒƒŠ‹Š—

IDirectSoundBuffer::GetVolume **IDirectSoundBuffer::SetVolume**

のの

IDirectSoundBuffer::GetFrequency
IDirectSoundBuffer::SetFrequency

の

の **IDirectSoundBuffer::GetPan**
IDirectSoundBuffer::SetPan

□•‚Ž“

IDirectSoundBuffer::GetCaps DirectSoundBuffer
IDirectSoundBuffer::GetStatus

IDirectSoundBuffer::GetFormat
の **IDirectSoundBuffer::GetFormat**
IDirectSoundBuffer::SetFormat

Note *ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ‚* ‚‚‚‚ ‚‚ŒŽ‚Œ’‚‚ ‚‚•ŒŽ‚*ƒƒƒƒƒ ƒƒƒƒ*‚•—
‚‚ ‚‚*ƒƒ ƒƒƒ‚* ‹‚*ƒƒƒƒ‚* ‚‚•—‚‚

ƒƒƒŠ—

IDirectSoundBuffer::Restore DirectSoundBuffer

IDirectSoundBuffer::Restore のの
のの

DirectSound‚Ž——

DirectSound の

1 **DirectSoundCreate**Š ‚Œ‚ ‚‚DirectSound*ƒƒƒƒƒƒ‚* ‚‚

2

IDirectSound::SetCooperativeLevel*ƒƒƒƒ*‚Œ‚ ‚‚‹’*ƒƒƒ*‚Ž’‚ ‚‚‚‚‚*ƒƒƒƒ ƒƒƒ*
‚‚ ‰ˆ*ƒƒƒ*‚‚‚DSSCL_NORMAL‚Ž—‚‚

3 **IDirectSound::CreateSoundBuffer***ƒƒƒƒ*‚Ž—
‚‚*ƒƒƒƒƒ ƒƒƒƒ‚* ‚‚ **DSBUFFERDESC** ‘‚’‚‚ ‚‚*ƒƒƒƒ*‚*ƒƒƒƒƒ ƒƒƒƒ*‚‚‚‚‚
Ž’‚‚•—‚‚‚ *ƒƒƒƒƒ*‚*ƒƒƒƒƒ ƒƒƒƒ‚* ‚‚‚‚‚‚‚‚‚‚

4 *ƒƒƒƒƒ ƒƒƒƒ*‚*ƒ ƒ*‚“‚ ‚ *ƒ ƒ*—
ˆ‚*ƒƒƒƒ*‚Ž“‚‚‚‚**IDirectSoundBuffer::Lock***ƒƒƒƒ*‚Ž—
‚ ‚‚*ƒƒƒƒ*‚*ƒ ƒ*‚*ƒƒƒ*‚‚‚‚**IDirectSoundBuffer::Unlock***ƒƒƒƒ*‚Ž—‚‚

5 **IDirectSoundBuffer::Play***ƒƒƒƒ*‚Ž—‚‚*ƒƒƒƒƒ ƒƒƒƒ‚* ‚‚

6 ƒƒƒƒ ƒƒƒ,ƒƒƒƒ, , —
,,, DirectSoundBufferƒƒƒƒ,**IDirectSoundBuffer::Stop**ƒƒƒƒ,Ž—
,,,,,,ƒƒƒƒ,’Ž,,,

7 ƒƒƒƒƒ ƒƒƒƒ,‰•,,

8 DirectSoundƒƒƒƒƒƒ,‰•,,

の

-

ƒƒƒƒƒ ƒƒƒƒ, ,,**IDirectSoundBuffer::SetFormat**ƒƒƒƒ,Œ, , ƒƒƒƒƒ ƒƒ
ƒƒ ƒƒƒƒ, —
ŒŽ, ’,, ,, ’,‘, ƒƒƒƒ ƒƒƒ,‹’ƒƒƒ,DSSCL_PRIORITY, ’,,,,,•—,,,

- ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ, , **IDirectSoundBuffer::Play**ƒƒƒƒ,Ž—
,,,,ƒƒƒƒ, ,, ,,, ,,,ƒƒƒƒƒ ƒƒƒƒ, ,,,,,,,, ƒƒƒƒƒ ƒƒƒƒ,,,, ,,,,,,,,•
,, ,,“ ,CPUƒƒ ,,,,, ”,,, ƒƒƒƒƒ ƒƒƒƒ, ,,,,,,ŠŽŽŠ,’ ,,

のDirectSound

- *DirectSoundƒƒƒƒƒƒ,*
- *CoCreateInstance,Ž—*,,DirectSoundƒƒƒƒƒƒ,
- *ƒ ƒƒƒƒ ”, ‰*
- *ƒƒƒƒ ƒƒƒƒ,*
- *ƒƒƒƒ ƒƒƒƒ,, ,,,*
- *DirectSoundƒƒƒ ,Ž—*
- *ƒƒƒƒ ƒƒƒ ,Ž—*
- *ˆ wave,Ž—*

DirectSoundƒƒƒƒƒƒ,□□

DirectSound **DirectSoundCreate**
NULL GUID
IDirectSound::SetCooperativeLevel

```
LPDIRECTSOUND lpDirectSound;
if(DS_OK == DirectSoundCreate(NULL, &lpDirectSound,
    NULL)) {
    // □□,□Œ,,□□
    lpDirectSound->lpVtbl->SetCooperativeLevel(lpDirectSound,
        hwnd, DSSCL_NORMAL);
    // .
    // . DirectSound,ƒƒƒƒ,,,,,ƒ□ƒ,,,,”’,,
    // .
} else {
    // □□,ž”,,□□
    // .
    // .
    // .
}
```

DirectSoundEnumerate

DSEnumCallback

```
typedef struct {
    // GUIDs,‹‰,,
    // ƒƒƒƒ,ƒƒƒƒƒƒƒƒƒ•Ž—,‹‰,,
} APPINSTANCEDATA, *LPAPPINSTANCEDATA;
BOOL AppEnumCallbackFunction(
    LPGUID lpGuid,
    LPTSTR lpstrDescription,
    LPTSTR lpstrModule,
    LPVOID lpContext)
{
    LPAPPINSTANCEDATA lpInstance = (LPAPPINSTANCEDATA)
    lpContext;
    // lpInstance‘‘,GUID,ƒƒ,,
    // Strcpy‹•Ž—,lpInstance‘‘,
    return TRUE; // —‹,‘,,
}
```

の DirectSound

```
AppInitDirectSound()
{
    APPINSTANCEDATA AppInstanceData;
    LPGUID lpGuid;
    LPDIRECTSOUND lpDirectSound;
    HRESULT hr;
    DirectSoundEnumerate(AppEnumCallbackFunction,
    &AppInstanceData);
    lpGuid = AppLetUserSelectDevice(&AppInstanceData);

    // DirectSoundCreate,ƒƒ,,,,
    // •’,ƒƒƒƒ,,•—,,,

    hr = DirectSoundCreate(lpGuid, &lpDirectSound, NULL);
    // .
    // .
    // .
}
```

lpGuid

DirectSoundCreate

の

CoCreateInstance,Ž— ,,DirectSound*ƒƒƒƒƒƒ*,

CoCreateInstance DirectSound

1 Š ,NULL,Ž’,, COM, Š‰,,

```
if (FAILED(CoInitialize(NULL)))
```

```
        return FALSE;
```

2 **DirectSoundCreate**Š ,Ž—,,,, **CoCreateInstance**,,, **IDirectSound::Initialize***ƒƒƒƒ*,Ž—,,DirectSound*ƒƒƒƒƒƒ*, ,,

```
dsrval = CoCreateInstance(&CLSID_DirectSound,
        NULL, &IID_IDirectSound, &lpds);
    if(!FAILED(dsrval))
        dsrval = IDirectSound_Initialize(lpds, NULL);
```

CLSID_DirectSound DirectSound
IID_IDirectSound DirectSound *lpds*
CoCreateInstance

DirectSound **IDirectSound::Initialize**
の **DirectSoundCreate** GUID (の NULL)
DirectSound **DirectSoundCreate**
DirectSound

の **CoUninitialize** COM の

```
CoUninitialize();
```

ƒ□ƒƒƒƒ□”,□‰

DirectSound DirectSound の

IDirectSound::GetCaps
ののの **DSCAPS** □□□

```
AppDetermineHardwareCaps(LPDIRECTSOUND lpDirectSound)
{
    DSCAPS dscaps;
    HRESULT hr;
    dscaps.dwSize = sizeof(DSCAPS);
    hr = lpDirectSound->lpVtbl->GetCaps(lpDirectSound,
    &dscaps);
    if(DS_OK == hr) {
        // □Œ,,,DSCAPS□`‘,•‰,,
        // .
        // .
        // .
    }
    // .
    // .
    // .
}
```

DSCAPS
の
の

の

のの **IDirectSound::GetCaps** Windows

ƒƒƒƒ□ƒƒƒƒ,□□

のののののの

- Š–“,ƒƒƒƒ ƒƒƒƒ,
- Œƒƒƒƒƒ
- ƒƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ,ƒƒƒ ƒ ƒƒƒƒ ƒƒƒƒ
- ƒ ƒƒƒƒ ƒƒƒƒ ƒƒƒƒ,ƒƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ
- ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ,ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ

Š–“,ƒƒƒƒ□ƒƒƒƒ,□□

DSBUFFERDESC
IDirectSound::CreateSoundBuffer
DirectSoundBuffer *IDirectSoundBuffer*

ののののDirectSound のの

のの

```
BOOL AppCreateBasicBuffer(
    LPDIRECTSOUND lpDirectSound,
    LPDIRECTSOUNDBUFFER *lplpDsb)
{
    PCMWAVEFORMAT pcmwf;
    DSBUFFERDESC dsbdesc;
    HRESULT hr;
    // waveƒƒ□ƒƒƒ□‘‘,□’,,
    memset(&pcmwf, 0, sizeof(PCMWAVEFORMAT));
    pcmwf.wf.wFormatTag = WAVE_FORMAT_PCM;
    pcmwf.wf.nChannels = 2;
    pcmwf.wf.nSamplesPerSec = 22050;
    pcmwf.wf.nBlockAlign = 4;
    pcmwf.wf.nAvgBytesPerSec =
        pcmwf.wf.nSamplesPerSec * pcmwf.wf.nBlockAlign;
    pcmwf.wBitsPerSample = 16;
    // DSBUFFERDESC□‘‘,□’,,
    memset(&dsbdesc, 0, sizeof(DSBUFFERDESC)); // 0,□’□
    dsbdesc.dwSize = sizeof(DSBUFFERDESC);
    // ƒƒƒƒƒ,ƒƒƒƒ□ƒ,–‹,,(ƒƒ□ƒƒƒ□ƒ□Ž”□)□
    dsbdesc.dwFlags = DSBCAPS_CTRLDEFAULT;
    // 3•,ƒƒƒƒ
    dsbdesc.dwBufferBytes = 3 * pcmwf.wf.nAvgBytesPerSec;
    dsbdesc.lpwfxFormat = (LPWAVEFORMATEX)&pcmwf;
    // ƒƒƒƒ,□□,,
    hr = lpDirectSound->lpVtbl->CreateSoundBuffer(lpDirectSound,
        &dsbdesc, lplpDsb, NULL);
    if(DS_OK == hr) {
        // □Œ□–Œ‚ƒƒƒ□ƒƒƒƒ‚*lplpDsb
        return TRUE;
    } else {
        // Ž”□
        *lplpDsb = NULL;
        return FALSE;
    }
}
```

□Œƒƒƒƒƒ

DSBUFFERDESC **dwFlags**
DirectSound

のの

IDirectSoundBuffer::SetVolume
のの DSBCAPS_CTRLVOLUME

DSERR_CONTROLUNAVAIL
のの

ƒƒƒƒƒƒ□ƒƒƒƒ□ƒƒƒƒ‚ƒƒƒ□ƒ□ƒƒƒƒ□ƒƒƒƒ

1 の
の 2 のの 3 ののの

DSBCAPS_STATIC

DirectSound

PCI のののの1 の

Note ƒƒƒƒ‚ƒƒƒƒƒƒ‚‚‚‚ƒƒƒ ƒ‚‚‚‚‚Ž’‚‚‚‚ƒƒƒ ƒƒƒ‚ “‰‚‚‚‚‚‚ ƒƒƒƒ‚‚‚‚‚‚Ž—
‚‚‚‚ Œ‚‚‚‚‚‚‚‚

ƒ□ƒƒƒƒ□ƒƒƒƒ□ƒƒƒƒ‚ƒƒƒƒƒƒ□ƒƒƒƒ□ƒƒƒƒ

の
CPU
の

DSBUFFERDESC
DSBCAPS_LOCHARDWARE DSBCAPS_LOCSOFTWARE
DSBCAPS_LOCHARDWARE

のの **IDirectSoundBuffer::GetCaps** **DSBCAPS** **dwFlags** DSBCAPS_LOCHARDWARE DSBCAPS_LOCSOFTWARE

ƒƒƒƒƒ□ƒƒƒƒ□ƒƒƒƒ‚ƒƒƒƒƒ□ƒƒƒƒ□ƒƒƒƒ

DSBUFFERDESC

DirectSound のののの

の **IDirectSoundBuffer::SetFormat** **IDirectSoundBuffer::Lock** **IDirectSoundBuffer::GetCurrentPosition** の

DirectSound の

DSBPLAY_LOOPING

のの

```
BOOL AppCreateWritePrimaryBuffer(
    LPDIRECTSOUND lpDirectSound, LPDIRECTSOUNDBUFFER *lplpDsb,
    LPDWORD lpdwBufferSize, HWND hwnd)
{
    DSBUFFERDESC dsbdesc;
    DSBCAPS dsbcaps;
    HRESULT hr;
    // wave format,□``,□',,
    memset(&pcmwf, 0, sizeof(PCMWAVEFORMAT));
    pcmwf.wf.wFormatTag = WAVE_FORMAT_PCM;
    pcmwf.wf.nChannels = 2;
    pcmwf.wf.nSamplesPerSec = 22050;
    pcmwf.wf.nBlockAlign = 4;
    pcmwf.wf.nAvgBytesPerSec =
        pcmwf.wf.nSamplesPerSec * pcmwf.wf.nBlockAlign;
    pcmwf.wBitsPerSample = 16;
    // DSBUFFERDESC□``,□',,
    memset(&lplpDsb, 0, sizeof(DSBUFFERDESC)); // 0,□'
    dsbdesc.dwSize = sizeof(DSBUFFERDESC);
    dsbdesc.dwFlags = DSBCAPS_PRIMARYBUFFER;
    // ƒƒƒƒƒƒƒ,ƒƒƒƒ‹Š,,,□Œ,,,
    dsbdesc.dwBufferBytes = 0;
    dsbdesc.lpwfxFormat = NULL; // ƒƒƒƒƒ□ƒƒƒƒ,NULL,,,,,,,,,

    // □,□,—□‹'ƒƒƒ,Ž"
    hr = lpDirectSound->lpVtbl->SetCooperativeLevel(lpDirectSound,
        hwnd, DSSCL_WRITEPRIMARY);
    if(DS_OK == hr) {
        // □Œ□ƒƒƒ,□□,Ž,,
        hr = lpDirectSound->lpVtbl->CreateSoundBuffer(lpDirectSound,
            &dsbdesc, lplpDsb, NULL);
        if(DS_OK == hr) {
            // □Œ□ƒƒƒƒ□ƒƒƒ,¬‹,,,ŒŽ,□’
            hr = (*lplpDsb)->lpVtbl->SetFormat(*lplpDsb, &pcmwf);
            if(DS_OK == hr) {
                // ƒƒƒƒƒƒ,’,,,,,,GetCaps,Œ,□,
                    dsbcaps.dwSize = sizeof(DSBCAPS);
                (*lplpDsb)->lpVtbl->GetCaps(*lplpDsb, &dsbcaps);
                *lpdwBufferSize = dsbcaps.dwBufferBytes;
                return TRUE;
            }
        }
    }
    // SetCooperativeLevel,Ž”
    // CreateSoundBuffer□,,,,SetFormat
    *lplpDsb = NULL;
    *lpdwBufferSize = 0;
    return FALSE;
}
```

ƒƒƒƒ□ƒƒƒƒ,,□,,,

の IDirectSoundBuffer::Lock ( ) のの

DirectSound 400

の

の

PCMCIA の

の

の □□□□□□ □□□□ DirectSound のの

の **IDirectSoundBuffer::Lock** **IDirectSoundBuffer::Unlock**

```
BOOL AppWriteDataToBuffer(
    LPDIRECTSOUNDBUFFER lpDsb, DWORD dwOffset, LPBYTE lpbSoundData,
    DWORD dwSoundBytes)
{
    LPVOID lpvPtr1;
    DWORD dwBytes1;
    LPVOID lpvPtr2;
    DWORD dwBytes2;
    HRESULT hr;
    // ƒƒƒƒƒƒ,Ž“
    hr = lpDsb->lpVtbl->Lock(lpDsb, dwOffset, dwSoundBytes, &lpvPtr1,
        &dwBytes1, &lpvPtr2, &dwBytes2, 0);
    // BUFFERLOST,•,,,□□□ƒƒƒƒ,□ƒƒƒ,□,
    if(DSERR_BUFFERLOST == hr) {
        lpDsb->lpVtbl->Restore(lpDsb);
        hr = lpDsb->lpVtbl->Lock(lpDsb, dwOffset, dwSoundBytes,
            &lpvPtr1, &dwAudio1, &lpvPtr2, &dwAudio2, 0);
    }
    if(DS_OK == hr) {
        // ƒƒƒƒ,□,,,,□,
        CopyMemory(lpvPtr1, lpbSoundData, dwBytes1);
        if(NULL != lpvPtr2) {
            CopyMemory(lpvPtr2, lpbSoundData+dwBytes1, dwBytes2);
        }
        // DirectSound,ƒ□ƒ,•,
        hr = lpDsb->lpVtbl->Unlock(lpDsb, lpvPtr1, dwBytes1, lpvPtr2,
            dwBytes2);
        if(DS_OK == hr) {
            // □Œ□
            return TRUE;
        }
    }
    // ƒƒƒ□ƒƒƒ‰□□,,,,—‹,Ž”
    return FALSE;
}
```

DirectSoundƒƒƒ□,Ž—

DirectSound のの の *IDirectSoundBuffer* の
IDirectSoundBuffer::Lock **IDirectSoundBuffer::Unlock**
□□□ **IDirectSoundBuffer::Play**
IDirectSoundBuffer::Stop

IDirectSoundBuffer::Play
の
の 0

IDirectSoundBuffer::SetCurrentPosition

IDirectSoundBuffer::Play の
のの (
のの)
IDirectSoundBuffer::Play

□□□ Win32 **SetTimer**
SetEvent
DirectSound DirectDraw

DirectSound

DSBCAPS_STATIC
DirectSound の

DirectSound のの
の

wave のの
DirectSound のの

の **IDirectSoundBuffer::SetFormat** □□□□□□□□
□□ の の
DSSCL_PRIORITY ののの DirectSound **IDirectSoundBuffer::SetFormat**

ƒƒƒƒ□ƒƒƒ□‚Ž—

の DirectSound のの
DirectSound
□
□□
のののののの

DSSCL_WRITEPRIMARY
IDirectSoundBuffer::Lock
□□□
IDirectSoundBuffer::Unlock DirectSound
IDirectSoundBuffer::Play □□□□□□□□□□□
DSBPLAY_LOOPING
IDirectSoundBuffer::Play のの

のの AppMixIntoPrimaryBuffer
のCustomMixer
CustomMixer のの□□□

```
BOOL AppMixIntoPrimaryBuffer(
    LPAPPSTREAMINFO lpAppStreamInfo, LPDIRECTSOUNDBUFFER
lpDsbPrimary,
    DWORD dwDataBytes, DWORD dwOldPos, LPDWORD lpdwNewPos)
{
    LPVOID lpvPtr1;
    DWORD dwBytes1;
    LPVOID lpvPtr2;
    DWORD dwBytes2;
    HRESULT hr;
    // □‚‚‚ƒƒƒƒ‚Ž‟
    hr = lpDsbPrimary->lpVtbl->Lock(lpDsbPrimary, dwOldPos,
dwDataBytes,
        &lpvPtr1, &dwBytes1, &lpvPtr2, &dwBytes2, 0);
    // DSERR_BUFFERLOST‚•‚‚‚‚□□‚□ƒƒƒƒ‚‚□ƒƒƒ‚‚‚‚
    if(DSERR_BUFFERLOST == hr) {
        lpDsbPrimary->lpVtbl->Restore(lpDsbPrimary);
        hr = lpDsbPrimary->lpVtbl->Lock(lpDsbPrimary, dwOldPos,
            dwDataBytes, &lpvPtr1, &dwBytes1, &lpvPtr2, &dwBytes2,
0);
```

```
    }
    if(DS_OK == hr) {
        // •,,,ƒƒƒƒ,ƒ□ƒ,ƒƒƒƒƒ,,
        CustomMixer(lpAppStreamInfo, lpvPtr1, dwBytes1);
        *lpdwNewPos = dwOldPos + dwBytes1;
        if(NULL != lpvPtr2) {
            CustomMixer(lpAppStreamInfo, lpvPtr2, dwBytes2);
            *lpdwNewPos = dwBytes2; // Š,−,,,,
        }
        // DirectSound,ƒ□ƒ,•,
        hr = lpDsbPrimary->lpVtbl->Unlock(lpDsbPrimary, lpvPtr1,
            dwBytes1, lpvPtr2, dwBytes2);
        if(DS_OK == hr) {
            // □Œ
            return TRUE;
        }
    }
    // ƒƒƒ,,,ƒƒƒ‰□,Ž”
    return FALSE;
}
```

ˆ□wave,Ž—

の DirectSound wave
PCM(pulse-coded modulation)
Win32 SDK (ACM)
の ACM

DirectSound*ƒƒƒƒƒƒ*

Š□

DirectSoundCreate

```
HRESULT DirectSoundCreate(GUID FAR * lpGuid,
    LPDIRECTSOUND * ppDS, IUuknown FAR * pUnkOuter);
```

IDirectSound の)

· DS_OK の)

DSERR_ALLOCATED

DSERR_INVALIDPARAM

DSERR_NOAGGREGATION

DSERR_NODRIVER

DSERR_OUTOFMEMORY

lpGuid
ƒƒƒƒƒƒƒƒ,Ž•,,GUID,,ƒƒƒƒ□
,,ƒƒƒ ƒ,’,**DirectSoundEnumerate**,,,•,,,GUIDs,,,, ,,,,ƒƒƒƒƒ,ƒƒƒƒ,—‹,,,,,
NULL,,,

ppDS
,,Š ,‰“,, ,,,DirectSoundƒƒƒƒƒƒ,,ƒƒƒƒ

pUnkOuter
–’‹,ˆ ,,’,NULL,,,,,,,,,

DirectSound **IDirectSound::SetCooperativeLevel**

IDirectSound::GetCaps, **IDirectSound::SetCooperativeLevel**

DirectSoundEnumerate

```
BOOL DirectSoundEnumerate(
    LPDSENUMCALLBACK lpDSEnumCallback, LPVOID lpContext);
```

DirectSound

- Œ,,,,,_,, Ž”,,,**DSERR_INVALIDPARAM**,ƒƒ ƒ ƒ,•,

lpDSEnumCallback
DSEnumCallbackŠ ,,ƒƒƒƒ ,,,ƒƒƒƒ,ƒƒƒƒ ƒ,,,,,,,,,,DirectSoundƒƒƒƒƒƒ,,
Œ, ,,,

lpContext
ƒ ƒ’‹ƒƒƒƒƒƒ,,ƒƒƒƒ ,,ƒƒƒƒƒƒ, —‹ƒ ƒƒƒƒŠ ,Œ, ,,,,,, ,,Š ,“,,,

DSEnumCallback

ƒ□ƒƒƒƒŠ□

DSEnumCallback

```
BOOL DSEnumCallback(GUID FAR * lpGuid,
    LPSTR lpstrDescription, LPSTR lpstrModule,
    LPVOID lpContext);
```

DirectSound

- ƒƒƒƒ‚—‹‚‘‚ ‚‚TRUE —‚‚ ‚‚FALSE‚•‚

lpGuid
—‹’‚ƒƒƒƒ‚Ž•‚‚GUID‚‚ƒƒƒƒ ‚‚’‚ ƒƒƒƒ‚DirectSoundƒƒƒƒƒƒ‚ ‚‚‚‚‚**DirectSoundCreate**Š ‚“‚‚‚

lpstrDescription
DirectSoundƒƒƒƒ‚ƒƒƒƒ‹ ‚—‚‚ NULL‚ ‚‚•Ž—‚‚ƒƒƒƒ

lpstrModule
‚‚ƒƒƒƒ‚‘“‚‚DirectSoundƒƒƒƒ‚ƒƒƒ ƒ–‚Ž’‚‚ NULL‚ ‚‚•Ž—‚‚ƒƒƒƒ

lpContext
ƒƒƒƒ ƒƒƒ’‹ƒ ƒ‚‚ƒƒƒƒ ‚‚‚‚‚ƒ ƒƒƒƒŠ ‚“‚‚‚

lpstrDescription *lpstrModule* ののののの

DirectSoundEnumerate

IDirectSound

DirectSound **IDirectSound** ののののの *IDirectSound*

IDirectSound のの

ƒƒƒŠ‚“‚	**Compact** **Initialize**
ƒƒƒƒ	**CreateSoundBuffer** **DuplicateSoundBuffer** **SetCooperativeLevel**
ƒƒƒƒ ”	**GetCaps**
ƒƒ ƒ ’	**GetSpeakerConfig** **SetSpeakerConfig**

の COM **IDirectSound** *IUuknown* **IUuknown** のの

AddRef

QueryInterface

Release

IDirectSound::Compact

```
HRESULT Compact();
```

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_INVALIDPARAM

  DSERR_PRIOLEVELNEEDED

  DSERR_UNINITIALIZED

の DirectSound (**IDirectSound::SetCooperativeLevel** DSSCL_EXCLUSIVE)のの

IDirectSound, **IDirectSound::SetCooperativeLevel**

IDirectSound::CreateSoundBuffer

```
HRESULT CreateSoundBuffer(LPDSBUFFERDESC lpDSBufferDesc,
    LPLPDIRECTSOUNDBUFFER lplpDirectSoundBuffer,
    IUuknown FAR * pUnkOuter);
```

DirectSoundBuffer

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_ALLOCATED

  DSERR_BADFORMAT

  DSERR_INVALIDPARAM

  DSERR_NOAGGREGATION

  DSERR_OUTOFMEMORY

  DSERR_UNINITIALIZED

  DSERR_UNSUPPORTED

lpDSBufferDesc
,,,ƒƒƒƒ ƒƒƒƒ,‹ ,Š,,,,,**DSBUFFERDESC** ‘‘,,ƒƒƒƒ

lplpDirectSoundBuffer
‹,DirectSoundBufferƒƒƒƒƒƒ,,ƒƒƒƒ ƒƒƒƒ, ,,,,,, ,NULL,,,

pUnkOuter
–’‹,ˆ ,,’,NULL,,,,,,,,,

IDirectSound::SetCooperativeLevel DirectSound

lpDSBufferDesc の DirectSound
DSBCAPS_CTRLFREQUENCY
IDirectSoundBuffer::SetFrequency の

ののの
DSBCAPS_STATIC
のDSBCAPS_LOCHARDWARE DSBCAPS_LOCSOFTWARE

DSBUFFERDESC, *IDirectSound*,

IDirectSound::DuplicateSoundBuffer, **IDirectSound::SetCooperativeLevel**, *IDirectSoundBuffer*, **IDirectSoundBuffer::GetFormat**, **IDirectSoundBuffer::GetVolume**, **IDirectSoundBuffer::Lock**, **IDirectSoundBuffer::Play**, **IDirectSoundBuffer::SetFormat**, **IDirectSoundBuffer::SetFrequency**

IDirectSound::DuplicateSoundBuffer

```
HRESULT DuplicateSoundBuffer(
    LPDIRECTSOUNDBUFFER lpDsbOriginal,
    LPLPDIRECTSOUNDBUFFER lplpDsbDuplicate);
```

DirectSoundBuffer

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_ALLOCATED

  DSERR_INVALIDCALL

  DSERR_INVALIDPARAM

  DSERR_OUTOFMEMORY

  DSERR_UNINITIALIZED

lpDsbOriginal
- ,,,DirectSoundBufferƒƒƒƒƒƒ,,ƒƒƒƒ

lplpDsbDuplicate
‹,DirectSoundƒƒƒƒƒƒ,,ƒƒƒƒ

の

のの

の

IDirectSound, **IDirectSound::CreateSoundBuffer**

IDirectSound::GetCaps

```
HRESULT GetCaps(LPDSCAPS lpDSCaps);
```

DirectSound

- Œ,,,DS_OK   Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_GENERIC

  DSERR_INVALIDPARAM

  DSERR_UNINITIALIZED

lpDSCaps
ƒƒƒƒ ƒƒƒƒ, ”,Ž,,,,,**DSCAPS** ‘‘,,ƒƒƒƒ

DSCAPS
の

DirectSoundCreate, **DSCAPS**, *IDirectSound*

IDirectSound::GetSpeakerConfig

```
HRESULT GetSpeakerConfig(LPDWORD lpdwSpeakerConfig);
```

DirectSound

- Œ,,,DS_OK   Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_INVALIDPARAM

  DSERR_UNINITIALIZED

lpdwSpeakerConfig
,,DirectSoundƒƒƒƒƒƒ,ƒƒ ƒ ’,,ƒƒƒƒ ƒƒ ƒ ’,ˆ‰,’,Ž’,,,

DSSPEAKER_HEADPHONE

ƒ ƒƒƒ,ƒƒƒƒƒ,, —,,

DSSPEAKER_MONO

ƒ ƒƒƒ,ƒƒƒƒ ƒƒ ƒ,, —,,

DSSPEAKER_QUAD

ƒ ƒƒƒ,4ƒƒƒƒƒ ƒƒ ƒ,, —,,

DSSPEAKER_STEREO

ƒ ƒƒƒ,ƒƒƒƒ ƒƒ ƒ,, —,,,(ƒƒƒƒƒ’)

DSSPEAKER_SURROUND

ƒ ƒƒƒ,ƒƒƒƒƒ ƒƒ ƒ,, —,,,

IDirectSound, **IDirectSound::SetSpeakerConfig**

IDirectSound::Initialize

```
HRESULT Initialize(GUID FAR * lpGuid);
```

CoCreateInstance DirectSound

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_ALREADYINITIALIZED

  DSERR_GENERIC

  DSERR_INVALIDPARAM

  DSERR_NODRIVER

lpGuid

ƒƒ ƒƒƒƒ ƒŽ•Ž(GUID)
,,ƒƒƒƒ ,,Ž•Ž, ,,DirectSoundƒƒƒƒƒƒ,Œ ,,,,,ƒƒƒƒƒƒƒƒ,Ž’,, NULL, ,ƒ
ƒƒƒƒ ƒƒƒƒ ƒƒƒƒ,‘‘,,,

の COM DirectSound **DirectSoundCreate**
の DSERR_ALREADYINITIALIZED **CoCreateInstance**
DirectSound **IDirectSound::Initialize**
の **DSERR_UNINITIALIZED**

CoCreateInstance **IDirectSound::Initialize** の *CoCreateInstance*
DirectSound の

DirectSoundCreate

IDirectSound::SetCooperativeLevel

```
HRESULT SetCooperativeLevel(HWND hwnd, DWORD dwLevel);
```

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_ALLOCATED

  DSERR_INVALIDPARAM

  DSERR_UNINITIALIZED

  DSERR_UNSUPPORTED

hwnd

ƒƒƒƒ ƒƒƒ,ƒƒƒƒƒ ƒƒƒƒ

dwLevel

—‹,,,‹’ƒƒƒ Ž,’,Ž’,,,

DSSCL_EXCLUSIVE

ƒƒƒƒ ƒƒƒ,”‘ƒƒƒ, ’,, “—ƒƒ ƒƒ,“,, ,,ƒƒƒƒ ƒƒƒ,,,—ˆ–
,,,,,,,, (DSBCAPS_GLOBALFOCUSƒƒƒ ’,ƒƒƒƒ ƒƒƒ,,,ƒƒƒƒ,ƒƒ ƒ,,,)
,,ƒƒƒ,,,, ƒƒƒƒ ƒƒƒ, DSSCL_PRIORITYƒƒƒ,,,,,“Œ,—
, ,**IDirectSoundBuffer::SetFormat**ƒƒƒƒ,Œ, ,,Ž’,,,, DirectSoun

d,ƒ ƒƒƒƒŒŽ,ƒƒƒƒ,, (,,, DirectSound, ’,,,—
ƒƒƒ,Š,,, ,,,wave,ƒƒƒƒ,,)

DSSCL_NORMAL

ƒƒƒƒ ƒƒƒ, •,‹’ ‘, ’,, ,ƒƒ ƒ,ƒƒƒƒƒƒƒƒ,ƒƒ ƒ‹—
, ,,, ,,,,,ƒƒƒƒ ƒƒƒ,,,ƒƒƒ, ’,,,

DSSCL_PRIORITY

ƒƒƒƒ ƒƒƒ,—
ƒƒƒ, ’,, ,,ƒƒƒ,ƒƒƒƒ ƒƒƒ,**IdirectSoundBuffer::SetFormat**,,,**IDirectSound::Compact**ƒƒƒƒ,Œ, ,,,,,,,

DSSCL_WRITEPRIMARY

,,, ˆ,ƒƒƒ,,, ƒƒƒƒ ƒƒƒ,ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ,, ,,,ƒƒƒƒ,—
,,,, ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ, ,ˆ ,,,,

の
の DSSCL_NORMAL の

IDirectSound, **IDirectSound::Compact**, **IDirectSoundBuffer::GetFormat**, **IDirectSoundBuffer::GetVolume**, **IDirectSoundBuffer::Lock**, **IDirectSoundBuffer::Play**, **IDirectSoundBuffer::Restore**, **IDirectSoundBuffer::SetFormat**

IDirectSound::SetSpeakerConfig

```
HRESULT SetSpeakerConfig(DWORD dwSpeakerConfig);
```

DirectSound

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

DSERR_INVALIDPARAM

DSERR_UNINITIALIZED

dwSpeakerConfig
DirectSoundƒƒƒƒƒƒ,ƒƒ ƒ ’ ˆ‰,’,Ž’,,,

DSSPEAKER_HEADPHONE

ƒƒ ƒ,ƒƒƒƒƒ,,,

DSSPEAKER_MONO

ƒƒ ƒ,ƒƒƒƒ,,,

DSSPEAKER_QUAD

ƒƒ ƒ,4ƒƒƒƒƒ,,,

DSSPEAKER_STEREO

ƒƒ ƒ,ƒƒƒƒ,,,(ƒƒƒƒƒ’)

DSSPEAKER_SURROUND

ƒƒ ƒ,ƒƒƒƒƒ ƒƒƒƒ,,,

IDirectSound, **IDirectSound::GetSpeakerConfig**

IDirectSound3DBuffer

の 3D の
IDirectSound3DBuffer のののののの
IDirectSound3DBuffer

IDirectSound3DBuffer のの

ffffff f ‘	**GetAllParameters** **SetAllParameters**
‹—	**GetMaxDistance** **GetMinDistance** **SetMaxDistance** **SetMinDistance**
‘ *f f*	**GetMode** **SetMode**
ˆ’	**GetPosition** **SetPosition**
ffff ffffffff *f f*	**GetConeAngles** **GetConeOrientation** **GetConeOutsideVolume** **SetConeAngles** **SetConeOrientation** **SetConeOutsideVolume**
‘“	**GetVelocity** **SetVelocity**

の COM **IDirectSound3DBuffer**
IUuknown **IUuknown** のの

AddRef

QueryInterface

Release

IDirectSound3DBuffer::GetAllParameters

```
HRESULT GetAllParameters(LPDS3DBUFFER lpDs3dBuffer);
```

のの 3D

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

lpDs3dBuffer
DS3DBUFFER ‘‘,,ƒƒƒƒ ,, ‘‘,ƒƒƒƒ ƒƒƒƒ,3D“ ,,,,, •,Ž,,,,

IDirectSound3DBuffer::GetConeAngles

```
HRESULT GetConeAngles(
    LPDWORD lpdwInsideConeAngle, LPDWORD lpdwOutsideConeAngle);
```

のの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM** ƒƒ ƒ ƒ,•,

lpdwInsideConeAngle *lpdwOutsideConeAngle*
ƒƒƒƒ ƒƒƒƒƒƒƒƒ ƒ ƒ,“•Š,Š•Š,Ž,• ,,ƒƒƒƒ

IDirectSound3DBuffer::GetConeOrientation

```
HRESULT GetConeOrientation(LPD3DVECTOR lpvOrientation);
```

のの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

lpvOrientation
D3DVECTOR ‘‘,,ƒƒƒƒ ,, ‘‘,ƒƒƒƒ ƒƒƒƒƒƒƒƒ ƒ ƒ,Œ ,•Œ,Ž,,,, ƒƒƒ ƒ •,ƒƒƒƒ ƒ ƒ,’ ,•,

IDirectSound3DBuffer::SetConeAngles, **IDirectSound3DBuffer::SetConeOutsideVolume**

IDirectSound3DBuffer::GetConeOutsideVolume

```
HRESULT GetConeOutsideVolume(LPLONG lplConeOutsideVolume);
```

の

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

lplConeOutsideVolume
ƒƒƒƒ,Œ ,ƒ ƒŠ•ƒƒƒ ƒ,Ž,• ,,ƒƒƒƒ

0( ) -10,000( )の DirectSound

の

IDirectSoundBuffer::SetVolume

IDirectSound3DBuffer::GetMaxDistance

```
HRESULT GetMaxDistance(LPD3DVALUE lpflMaxDistance);
```

ののの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM** *ƒƒ ƒ ƒ*,•,

lpflMaxDistance

Œ , ’‹— ’,Ž,• ,,*ƒƒƒƒ*

の

□□□□□□□□□□□□□□□□□□□

IDirectSound3DBuffer::GetMinDistance, **IDirectSound3DBuffer::SetMaxDistance**

IDirectSound3DBuffer::GetMinDistance

```
HRESULT GetMinDistance(LPD3DVALUE lpflMinDistance);
```

ののの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ƒƒ ƒ ƒ*,•,

lpflMinDistance

Œ , ’‹— ’,Ž,• ,,*ƒƒƒƒ*

1.0 ののの

1.0 の

□□□□□□□□□□□□□□□□□□□

IDirectSound3DBuffer::SetMinDistance, **IDirectSound3DBuffer::GetMaxDistance**

IDirectSound3DBuffer::GetMode

```
HRESULT GetMode(LPDWORD lpdwMode);
```

3D ののの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ƒƒ ƒ ƒ*,•,

lpdwMode

Œ ,‘ *ƒ ƒ*,Ž,• ,,*ƒƒƒƒ* Ž,’,•,

DS3DMODE_DISABLE

3D*ƒƒƒƒ* —,–Œ,,, *ƒƒƒƒ*,*ƒƒƒ* ,“,’ ,,•,,,,,,Š,,

DS3DMODE_HEADRELATIVE

ƒƒƒƒƒƒƒ ƒ(ˆ’ ‘“ •Œ)
,*ƒƒƒ* ,*ƒƒƒ ƒ*,‘‘ŠŒ,,, ,,*ƒ ƒ*,, *ƒƒƒƒ*, ‘*ƒƒƒ ƒ*,*ƒƒƒ ƒƒƒ ƒ*,• ,,,,,,
Ž““, ,,,,, ‘‘*ƒƒƒ ƒ*,,•‰,,,,,,,,

DS3DMODE_NORMAL

• ‚ — ‚‚‚ƒƒƒƒƒƒ ƒ‚‚‚‚‚

IDirectSound3DBuffer::GetPosition

```
HRESULT GetPosition(LPD3DVECTOR lpvPosition);
```

ϑϑ

- Œ‚‚‚DS_OK Ž”‚‚‚**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ‚•‚

lpvPosition
D3DVECTOR ‘‘‚‚ƒƒƒƒ ‚‚ ‘‘‚ƒƒƒƒ ƒƒƒƒ’‚ƒƒƒƒ ƒƒƒƒƒ‚Ž‚‚‚

IDirectSound3DBuffer::GetVelocity

```
HRESULT GetVelocity(LPD3DVECTOR lpvVelocity);
```

ϑϑ

- Œ‚‚‚DS_OK Ž”‚‚‚**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ‚•‚

lpvVelocity
D3DVECTOR ‘‘‚‚ƒƒƒƒ ‚‚ ‘‘‚ƒƒƒƒ ƒƒƒƒ‚Œ ‚‘“‚Ž‚‚‚
ϑ

IDirectSound3DBuffer::SetPosition,
IDirectSound3DBuffer::SetVelocity

IDirectSound3DBuffer::SetAllParameters

```
HRESULT SetAllParameters(
    LPDS3DBUFFER lpDs3dBuffer, DWORD dwApply);
```

ϑϑϑ **DS3DBUFFER**
ϑ 3D

- Œ‚‚‚DS_OK Ž”‚‚‚**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ‚•‚

lpDs3dBuffer
DS3DBUFFER ‘‘‚‚ƒƒƒƒ ‚‚ ‘‘‚ƒƒƒƒ ƒƒƒƒ‚3D“ ‚Ž‚‚‚

dwApply
’‚ƒƒƒƒƒ‚Ž‚’ Ž‚’‚Ž‚‚

DS3D_DEFERRED	ƒƒƒƒ ƒƒƒ‚**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ‚Œ‚ ‚‚‚ ’‚ ‚‚‚‚ • ‚ ’‚• ‚‚‚ ’“‚ ŒŽ‚Ž ‚‚‚‚‚‰”
DS3D_IMMEDIATE	’‚‘Ž‚ ‚‚ ƒƒƒƒ‚‚‚‚‚3Dƒƒƒƒ ƒƒƒƒ‚‘‚‚3D • ‚ ŒŽ‚ ‚

IDirectSound3DBuffer::SetConeAngles

```
HRESULT SetConeAngles(
    DWORD dwInsideConeAngle,
    DWORD dwOutsideConeAngle, DWORD dwApply);
```

ϘϪϘ

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ƒƒ ƒ ƒ*,•,

dwInsideConeAngle *dwOutsideConeAngle*
ƒƒƒƒ ƒƒƒƒƒƒƒƒ ƒ ƒ,“•Š,Š•Š

dwApply
’, ,,,*ƒƒƒƒƒ*,Ž,’ Ž,’,Ž,,,

DS3D_DEFERRED	*ƒƒƒƒ ƒƒƒ*,**IDirectSound3DListener::CommitDeferredSettings***ƒƒƒƒ*,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”
DS3D_IMMEDIATE	’,‘Ž, ,, *ƒƒƒƒ*,,,,,3D*ƒƒƒƒ ƒƒƒƒ*,‘,,3D • , ŒŽ, ,

Ϙ 0 ( ) 360 ( ) 360

IDirectSound3DBuffer::GetConeOutsideVolume, **IDirectSound3DBuffer::SetConeOutsideVolume**

IDirectSound3DBuffer::SetConeOrientation

```
HRESULT SetConeOrientation(D3DVALUE x,
    D3DVALUE y, D3DVALUE z, DWORD dwApply);
```

ϘϪϪϘ

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ƒƒ ƒ ƒ*,•,

x, y, *z*
D3DVALUE,*ƒƒƒ*,Ž,’ ,,, ‹,*ƒƒƒƒ ƒ ƒ*,•Œ*ƒƒƒƒ*, •,•,

dwApply
’, ,,,*ƒƒƒƒƒ*,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED	*ƒƒƒƒ ƒƒƒ*,**IDirectSound3Dlistener::CommitDeferredSettings***ƒƒƒƒ*,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”
DS3D_IMMEDIATE	’,‘Ž, ,, *ƒƒƒƒ*,,,,,3D*ƒƒƒƒ ƒƒƒƒ*,‘,,3D • , ŒŽ, ,

IDirectSound3DBuffer::GetConeOrientation *lpvOrientation* ϘϘ
(0,0,1)

IDirectSound3DBuffer::SetConeAngles, **IDirectSound3DBuffer::SetConeOutsideVolume**

IDirectSound3DBuffer::SetConeOutsideVolume

```
HRESULT SetConeOutsideVolume(
    LONG lConeOutsideVolume, DWORD dwApply);
```

のの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

lConeOutsideVolume
ƒƒƒƒ ƒƒƒƒ,ƒ ƒŠ•ƒƒƒ ƒ,100ƒƒƒƒ dB ,”ˆ,Ž’,, ”’‰”’”ˆ,0(Œ ,,) ,,-10,000(–‰),,,,,

dwApply
’, ,,,ƒƒƒƒƒ,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED	ƒƒƒƒ ƒƒƒ,**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”
DS3D_IMMEDIATE	’,‘Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,‘,,3D • , ŒŽ, ,

DirectSound

のの

IDirectSoundBuffer::SetVolume

IDirectSound3DBuffer::SetMaxDistance

```
HRESULT SetMaxDistance(
    D3DVALUE flMaxDistance, DWORD dwApply);
```

の

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

flMaxDistance
,, ’‹—’

dwApply
’, ,,,ƒƒƒƒƒ,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED	ƒƒƒƒ ƒƒƒ,**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”
DS3D_IMMEDIATE	’,‘Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,‘,,3D • , ŒŽ, ,

の

□□□□□□□□□□□□□□□□□□□

IDirectSound3DBuffer::GetMaxDistance, **IDirectSound3DBuffer::SetMinDistance**

IDirectSound3DBuffer::SetMinDistance

```
HRESULT SetMinDistance(
    D3DVALUE flMinDistance, DWORD dwApply);
```

の)

DS_OK **DSERR_INVALIDPARAM**

flMinDistance
,, ’‹—’

dwApply
’, ,,,ƒƒƒƒƒ,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED ƒƒƒƒ ƒƒƒ,**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”

DS3D_IMMEDIATE ’,‘Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,‘,,3D • , ŒŽ, ,

1.0(のの) 1.0
) の) □□□□□□□□□□□

IDirectSound3DBuffer::SetMaxDistance

IDirectSound3DBuffer::SetMode

```
HRESULT SetMode(
    DWORD dwMode, DWORD dwApply);
```

3D

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

dwMode
3Dƒƒƒƒ —ƒ ƒ, ’,, Ž,’,Ž,,,

DS3DMODE_DISABLE

3Dƒƒƒƒ —,–Œ ƒƒƒƒ,ƒƒƒ ,“,’ ,,•,,,,,,Š,,

DS3DMODE_HEADRELATIVE

ƒƒƒƒ ƒƒƒ ƒ(ˆ’ ‘“ •Œ)
,ƒƒƒ ,ƒƒƒ ƒ,‘‘ŠŒ,,, ,,ƒ ƒ,, ƒƒƒƒ, ‘ƒƒƒ ƒ,ƒƒƒ ƒƒƒ ƒ,• ,,,,,,
Ž““, ,,,,, ‘‘ƒƒƒ ƒ,•,,,,,,,,,

DS3DMODE_NORMAL

• , — ,,,ƒƒƒƒƒ ƒ ƒ,,,,,,

dwApply
’, ,,,ƒƒƒƒƒ,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED ƒƒƒƒ ƒƒƒ,**IdirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”

DS3D_IMMEDIATE ’,‘Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,‘,,3D • , ŒŽ, ,

IDirectSound3DBuffer::SetPosition

```
HRESULT SetPosition(D3DVALUE x,
    D3DVALUE y, D3DVALUE z, DWORD dwApply);
```

ᏍᏍᏍ

IDirectSound3DListener::SetDistanceFactor

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

x, y, z

D3DVALUE,ƒƒƒ,Ž,’ ,,, ,,ˆ’ƒƒƒƒ, •,•,

dwApply

’, ,,,ƒƒƒƒƒ,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED ƒƒƒƒ ƒƒƒ,**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”

DS3D_IMMEDIATE ’,‘Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,‘,,3D • , ŒŽ, ,

IDirectSound3DBuffer::SetVelocity

```
HRESULT SetVelocity(D3DVALUE x,
    D3DVALUE y, D3DVALUE z, DWORD dwApply);
```

ᏍᏍ

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

x, y, z

D3DVALUE,ƒƒƒ,Ž,’ ,,, ,,‘“ƒƒƒƒ, •,•,

dwApply

’, ,,,ƒƒƒƒƒ,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED ƒƒƒƒ ƒƒƒ,**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”

DS3D_IMMEDIATE ’,‘Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,‘,,3D • , ŒŽ, ,

Ꮝ

IDirectSound3DBuffer::SetPosition, **IDirectSound3DBuffer::GetVelocity**

IDirectSound3DListener

の 3D の
IDirectSound3DListener のののののIDirectSound3Dlistener*

IDirectSound3DListener のの

ffffff f‘	**GetAllParameters** **SetAllParameters**
,’, ’	**CommitDeferredSettings**
‹——‘	**GetDistanceFactor** **SetDistanceFactor**
ffff —‘	**GetDopplerFactor** **SetDopplerFactor**
•Œ	**GetOrientation** **SetOrientation**
ˆ’	**GetPosition** **SetPosition**
f fff—‘	**GetRolloffFactor** **SetRolloffFactor**
‘“	**GetVelocity** **SetVelocity**

の COM **IDirectSound3DBuffer**
IUuknown **IUuknown** のの

AddRef
QueryInterface
Release

IDirectSound3Dlistener ::CommitDeferredSettings

```
HRESULT CommitDeferredSettings();
```

ののの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

のの □□□□□□□□□□□□□□

IDirectSound3DListener::GetAllParameters

```
HRESULT GetAllParameters(LPDS3DLISTENER lpListener);
```

3D 3D のの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

lpListener
DS3DLISTENER ‘‘,,ƒƒƒƒ ,, ‘‘,3Dƒ ƒƒ,,,3Dƒƒƒ ,Œ , ‘,Ž,

DirectSound3DListener::SetAllParameters

IDirectSound3DListener::GetDistanceFactor

```
HRESULT GetDistanceFactor(LPD3DVALUE lpflDistanceFactor);
```

の

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

lpflDistanceFactor
D3DVALUEƒƒƒ,• ,,ƒƒƒƒ ,,,Œ ,‹——‘,’,Ž,,,

IDirectSound3DListener::SetDistanceFactor

IDirectSound3DListener::GetDopplerFactor

```
HRESULT GetDopplerFactor(LPD3DVALUE lpflDopplerFactor);
```

の

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

lpflDopplerFactor
D3DVALUEƒƒƒ,• ,,ƒƒƒƒ ,,,Œ ƒƒƒƒ —‘,’,Ž,,,

の 0( ) 10.0(のの 10 の)
DS3D_DEFAULTDOPPLERFACTOR (1.0)

IDirectSound3DListener::SetDopplerFactor

IDirectSound3DListener::GetOrientation

```
HRESULT GetOrientation(
    LPD3DVECTOR lpvOrientFront,
    LPD3DVECTOR lpvOrientTop);
```

のののの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ff f f*,•,

lpvOrientFront
D3DVECTOR ‘‘,,*ffff fff* ,‘•*ffff*,Ž,,,,

lpvOrientTop
D3DVECTOR ‘‘,,*ffff fff* , •*ffff*,Ž,,,,

(0,0,1.0) (0,1.0,0)

IDirectSound3DListener::SetOrientation

IDirectSound3DListener::GetPosition

```
HRESULT GetPosition(LPD3DVECTOR lpvPosition);
```

ののの**IDirectSound3DListener::SetDistanceFactor**

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ff f f*,•,

lpvPosition
D3DVECTOR ‘‘,,*ffff fff* ,Œ ,ˆ’,Ž,

IDirectSound3DListener::SetPosition

IDirectSound3DListener::GetRolloffFactor

```
HRESULT GetRolloffFactor(LPD3DVALUE lpflRolloffFactor);
```

の

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ff f f*,•,

lpflRolloffFactor
D3DVALUE*fff*,• ,,*ffff* Œ ,*f fff*—‘,’,Ž,,,,

DS3D_DEFAULTROLLOFFFACTOR (1.0)

IDirectSound3DListener::SetRolloffFactor

IDirectSound3DListener::GetVelocity

```
HRESULT GetVelocity(LPD3DVECTOR lpvVelocity);
```

のの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ƒƒ ƒ ƒ*,•,

lpvVelocity
D3DVECTOR ‘‘,,*ƒƒƒƒ* ,, ‘‘,*ƒƒƒƒ ƒƒƒƒ*,Œ ,‘“,Ž,,,,

のののの **IDirectSound3DListener::SetPosition**
(0,0,0)

IDirectSound3DListener::SetVelocity

IDirectSound3DListener::SetAllParameters

```
HRESULT SetAllParameters(
    LPDS3DLISTENER lpListener, DWORD dwApply);
```

DS3DBUFFER 3D ののの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ƒƒ ƒ ƒ*,•,

lpListener
DS3DBUFFER ‘‘,,*ƒƒƒƒ* Œ ,3D*ƒƒƒ ƒƒƒ ƒ* •,,,,Ž,,,,

dwApply
’, ,,,*ƒƒƒƒƒ*,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED	*ƒƒƒƒ ƒƒƒ*,**IDirectSound3Dlistener::CommitDeferredSettings***ƒƒƒƒ*,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,‰”,,,
DS3D_IMMEDIATE	’,‘Ž, ,, *ƒƒƒƒ*,,,,,3D*ƒƒƒƒ ƒƒƒƒ*,‘,,3D • , ŒŽ, ,

IDirectSound3DListener::GetAllParameters

IDirectSound3DListener::SetDistanceFactor

```
HRESULT SetDistanceFactor(
    D3DVALUE flDistanceFactor, DWORD dwApply);
```

の

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ƒƒ ƒ ƒ*,•,

flDistanceFactor
,,‹——‘

dwApply
’, ,,,*ƒƒƒƒƒ*,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED	*ƒƒƒƒ ƒƒƒ*,**IDirectSound3DListener::Com**

mitDeferredSettingsƒƒƒƒ,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”,,,

DS3D_IMMEDIATE

’,‘Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,‘,,3D • , ŒŽ, ,

IDirectSound3DListener::GetDistanceFactor

IDirectSound3DListener::SetDopplerFactor

```
HRESULT SetDopplerFactor(
    D3DVALUE flDopplerFactor, DWORD dwApply);
```

のの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

flDopplerFactor

,,ƒƒƒƒ —‘,’

dwApply

’, ,,,ƒƒƒƒƒ,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED

ƒƒƒƒ ƒƒƒ,**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,,,,,‰”,,,

DS3D_IMMEDIATE

’,‘Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,‘,,3D • , ŒŽ, ,

の 0( ) 10.0(のの 10 の))
DS3D_DEFAULTDOPPLERFACTOR (1.0)

IDirectSound3DListener::GetDopplerFactor

IDirectSound3DListener::SetOrientation

```
HRESULT SetOrientation(D3DVALUE xFront,
    D3DVALUE yFront, D3DVALUE zFront,
    D3DVALUE xTop, D3DVALUE yTop,
    D3DVALUE zTop, DWORD    dwApply);
```

のののの

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

xFront, yFront, zFront

D3DVALUEƒƒƒ,’ ‘•ƒƒƒƒ, •,Ž,,,,

xTop, yTop, zTop

D3DVALUEƒƒƒ,’ •ƒƒƒƒ, •,Ž,,,,

dwApply

’, ,,,ƒƒƒƒƒ,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED ƒƒƒƒ ƒƒƒ,**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ', ,,,, • , ',• ,,, '", ŒŽ,Ž ,,,,,‰",,,

DS3D_IMMEDIATE ','Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,',,3D • , ŒŽ, ,

(0,0,1.0) (0,1.0,0)

IDirectSound3DListener::GetOrientation

IDirectSound3DListener::SetPosition

```
HRESULT SetPosition(D3DVALUE x, D3DVALUE y,
    D3DVALUE z, DWORD dwApply);
```

ののの**IDirectSound3DListener::SetDistanceFactor**

- Œ,,,DS_OK Ž",,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

x, y, *z*

D3DVALUEƒƒƒ,' ƒƒƒ , ‹,ˆ'ƒƒƒƒ, •,Ž,,,,

dwApply

', ,,,ƒƒƒƒƒ,Ž,' ˆ‰,',Ž,,,

DS3D_DEFERRED ƒƒƒƒ ƒƒƒ,**IDirectSound3DListener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ', ,,,, • , ',• ,,, '", ŒŽ,Ž ,,,,,,

DS3D_IMMEDIATE ','Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,',,3D • , ŒŽ,,

IDirectSound3DListener::GetPosition

IDirectSound3DListener::SetRolloffFactor

```
HRESULT SetRolloffFactor(
    D3DVALUE flRolloffFactor, DWORD dwApply);
```

- Œ,,,DS_OK Ž",,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

flRolloffFactor

‹,ƒ ƒƒƒ—'

dwApply

', ,,,ƒƒƒƒƒ,Ž,' ˆ‰,',Ž,,,

DS3D_DEFERRED ƒƒƒƒ ƒƒƒ,**IdirectSound3Dlistener::CommitDeferredSettings**ƒƒƒƒ,Œ, ,,, ', ,,,, • , ',• ,,, '", ŒŽ,Ž ,‰",,,

DS3D_IMMEDIATE ','Ž, ,, ƒƒƒƒ,,,,,3Dƒƒƒƒ ƒƒƒƒ,',,3D •

, ŒŽ, ,

DS3D_DEFAULTROLLOFFFACTOR (1.0)

IDirectSound3DListener::GetRolloffFactor

IDirectSound3DListener::SetVelocity

```
HRESULT SetVelocity(D3DVALUE x,
    D3DVALUE y, D3DVALUE z, DWORD dwApply);
```

の

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ff f f*,•,

x, y, z
D3DVALUE*fff*,’ *fff* , ‹,‘“, •,Ž,,,,

dwApply
’, ,,,*fffff*,Ž,’ ˆ‰,’,Ž,,,

DS3D_DEFERRED	*ffff fff*,**IDirectSound3DListener::CommitDeferredSettings***ffff*,Œ, ,,, ’, ,,,, • , ’,• ,,, ’“, ŒŽ,Ž ,‰”,,,
DS3D_IMMEDIATE	’,‘Ž, ,, *ffff*,,,,,3D*ffff ffff*,‘,,3D • , ŒŽ, ,

のの **IDirectSound3DListener::SetPosition**
(0,0,0)

IDirectSound3DListener::GetVelocity

IDirectSoundBuffer

DirectSoundBuffer
IDirectSoundBuffer のの

•	**GetCaps**
	GetFormat
	GetStatus
	SetFormat
*fff*Š—	**Initialize**
	Restore
Š—	**GetCurrentPosition**
	Lock
	Play

SetCurrentPosition

Stop

Unlock

*ffff*Š— **GetFrequency**

GetPan

GetVolume

SetFrequency

SetPan

SetVolume

の COM **IDirectSound3DBuffer**
IUuknown
の

AddRef

QueryInterface

Release

IDirectSoundBuffer::GetCaps

```
HRESULT GetCaps(LPDSBCAPS lpDSBufferCaps);
```

DirectSoundBuffer

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM***ff f f*,•,

lpDSBufferCaps
DSBCAPS ‘‘,,*ffff ffff ffff*, ”,Ž,,,

DSBCAPS **IDirectSound::CreateSoundBuffer**
DSBUFFERDESC
ののののののののの

DSBCAPS **dwFlags** **DSBUFFERDESC**
の **DSBCAPS**
DSBCAPS_LOCHARDWARE DSBCAPS_LOCSOFTWARE の
のXの
DSBUFFERDESC

DSBCAPS, **DSBUFFERDESC**, *IDirectSoundBuffer*,
IDirectSound::CreateSoundBuffer

IDirectSoundBuffer::GetCurrentPosition

```
HRESULT GetCurrentPosition(LPDWORD lpdwCurrentPlayCursor,
    LPDWORD lpdwCurrentWriteCursor);
```

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_INVALIDPARAM

  DSERR_PRIOLEVELNEEDED

lpdwCurrentPlayCursor
DirectSoundBufferƒƒƒƒƒƒ“,Œ , ˆ’,Ž,• ,,ƒƒƒƒ ,,ˆ’,ƒƒƒƒ ƒƒƒƒ“,ƒƒƒƒƒ ,,, ƒƒƒ,Ž’,,,

lpdwCurrentWriteCursor
DirectSoundBufferƒƒƒƒƒƒ“,Œ , ,,,ˆ’,Ž,• ,,ƒƒƒƒ ,,ˆ’,ƒƒƒƒ ƒƒƒƒ“,ƒƒƒƒƒ ,,, ƒƒƒ,Ž’,,,

15 の

lpdwCurrentPlayCursor のの

IDirectSoundBuffer, **IDirectSoundBuffer::SetCurrentPosition**

IDirectSoundBuffer::GetFormat

```
HRESULT GetFormat(LPWAVEFORMATEX lpwfxFormat,
    DWORD dwSizeAllocated, LPDWORD lpdwSizeWritten);
```

- Œ,,,DS_OK Ž”,,,**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ,•,

lpwfxFormat
ƒƒƒƒ“,ƒƒƒƒ ƒ ƒ,,,,, •,Š,**WAVEFORMATEX** ‘‘,,ƒƒƒƒ ŒŽ, •,•— ,ƒƒƒƒ ƒƒƒ,Ž“,, ,NULL,Ž’,,

dwSizeAllocated

WAVEFORMATEX ‘‘,ƒƒƒ’ˆ,ƒƒƒ DirectSound,ƒƒƒƒ, ,,,,,,,,,*dwSizeAllocated*,ƒƒƒ ,,,,, ,,**WAVEFORMATEX** ‘‘,,,,‘,,ƒƒƒ,—‹,, , ,,,,,,

lpdwSizeWritten
WAVEFORMATEX ‘‘, ,,,,,ƒƒƒ ,Ž,• ,,ƒƒƒƒ ,,ƒƒƒ ƒ,NULL,,,,

WAVEFORMATEX

の DirectSoundBuffer のの

lpdwSizeWritten の

IDirectSoundBuffer, **IDirectSoundBuffer::SetFormat**

IDirectSoundBuffer::GetFrequency

```
HRESULT GetFrequency(LPDWORD lpdwFrequency);
```

の 1 の

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

DSERR_CONTROLUNAVAIL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

lpdwFrequency

ƒ ƒƒƒ ƒƒƒƒ‚ ‚‚‚‚‚ ‚Ž” ‚Ž‚• ‚‚ƒƒƒƒ

の 100 100,000 の

IDirectSoundBuffer, **IDirectSoundBuffer::SetFrequency**

IDirectSoundBuffer::GetPan

```
HRESULT GetPan(LPLONG lplPan);
```

- Œ‚‚‚DS_OK Ž”‚‚‚Ž‚ƒƒ ƒ ƒ‚•‚

DSERR_CONTROLUNAVAIL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

lplPan

‰‚ ‚‘ƒƒƒƒ‚Ž‚• ‚‚ƒƒƒƒ

-10,000 10,000 の 10,000 100dB -10,000 のの 100dB 10,000 のの 100dB の 0 *lplPan* 0 の(0dB) 0 の

-2173 の 21.73 dB 870 の 8.7 dB -10,000 のの10,000 のの

IDirectSoundBuffer, **IDirectSoundBuffer::GetVolume**, **IDirectSoundBuffer::SetPan**, **IDirectSoundBuffer::SetVolume**

IDirectSoundBuffer::GetStatus

```
HRESULT GetStatus(LPDWORD lpdwStatus);
```

のの

- Œ‚‚‚DS_OK Ž”‚‚‚**DSERR_INVALIDPARAM**ƒƒ ƒ ƒ‚•‚

lpdwStatus

ƒƒƒƒ ƒƒƒƒ‚ ‘‚Ž‚• ‚‚ƒƒƒƒ ‘‚Ž‚’‚ ’‚‚‚

DSBSTATUS_BUFFERLOST

ƒƒƒƒ‚Ž‚‚‚‚ ‚‚‚ƒƒƒ‚ ‚‘‚ƒƒƒƒ‚‚•—‚‚‚

DSBSTATUS_LOOPING

ƒƒƒƒ‚ƒ ƒ‚‚‚‚ ’‚–
’‚‚‚‚ ƒƒƒƒ‚ƒƒƒƒƒ ƒ‚ Œ‚ ‚’Ž‚‚ ‚‚’‚ ’Ž‚‚ ƒƒƒƒ‚ ‚ ‚‚‚‚‚‚‚‚’ˆ‚‚

DSBSTATUS_PLAYING

ƒƒƒƒ‚ ’‚‚‚ ‚‚’‚ ’‚‚‚‚‚‚‚ƒƒƒƒ‚’Ž‚‚‚‚

IDirectSoundBuffer

IDirectSoundBuffer::GetVolume

```
HRESULT GetVolume(LPLONG lplVolume);
```

のの

- Œ‚‚‚DS_OK Ž”‚‚‚Ž‚ƒƒ ƒ ƒ‚•‚

DSERR_CONTROLUNAVAIL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

lplVolume
Ž’‚‚‚DirectSoundƒƒƒƒ‚ƒƒƒ ƒ‚Ž‚• ‚‚ƒƒƒƒ

100dB の 0 -10,000 の0 ののの
-10,000 100 dB

の10dB の 20dB 4 の 1

IDirectSoundBuffer, **IDirectSoundBuffer::SetVolume**

IDirectSoundBuffer::Initialize

```
HRESULT Initialize(LPDIRECTSOUND lpDirectSound,
    LPDSBUFFERDESC lpDSBufferDesc);
```

DirectSoundBuffer

- **DSERR_ALREADYINITIALIZED**ƒƒ ƒ ƒ‚•‚

lpDirectSound
DirectSoundBufferƒƒƒƒƒƒ‚Œ ‚‚DirectSoundƒƒƒƒƒƒ‚‚ƒƒƒƒ

lpDSBufferDesc
ƒƒƒƒ ƒƒƒƒ‚ Š‰‚Ž—‚‚’‚Ž‚‚‚‚**DSBUFFERDESC** ‘‘‚‚ƒƒƒƒ

IDirectSound::CreateSoundBuffer
IDirectSoundBuffer::Initialize DirectSound の

DSBUFFERDESC, **IDirectSound::CreateSoundBuffer**, *IDirectSoundBuffer*

IDirectSoundBuffer::Lock

```
HRESULT Lock(DWORD dwWriteCursor, DWORD dwWriteBytes,
    LPVOID lplpvAudioPtr1, LPDWORD lpdwAudioBytes1,
    LPVOID lplpvAudioPtr2, LPDWORD lpdwAudioBytes2,
    DWORD dwFlags);
```

の

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_BUFFERLOST

  DSERR_INVALIDCALL

  DSERR_INVALIDPARAM

  DSERR_PRIOLEVELNEEDED

dwWriteCursor
ƒƒƒƒ,ŠŽˆ’,,ƒƒƒ,ŠŽ,,ˆ’,,, ƒƒƒ’ˆ,ƒƒƒƒƒ
*dwFlags*ƒƒƒ ƒ,DSBLOCK_FROMWRITECURSORƒƒƒ,Ž’,,,,, , ,,ƒƒƒ
ƒ,–Ž,,,

dwWriteBytes
ƒƒƒ,,ƒƒƒƒ—ˆ, ƒƒƒ’ˆ,ƒƒƒ ƒƒƒƒ ƒƒƒƒ,Š”“,,‰,,,,,, ƒƒƒƒƒƒƒ ,,,’ˆ,,,,

lplpvAudioPtr1
ƒƒƒ,,,ƒƒƒƒ, ,ƒƒƒƒ,,ƒƒƒƒ

lpdwAudioBytes1
*lplpvAudioPtr1*ƒƒƒ ƒ,Ž,,,ƒƒƒ ,Ž,• ,,ƒƒƒƒ ,,’,*dwWriteBytes*ƒƒƒ ƒ,, ,,
lplpvAudioPtr2,ƒƒƒƒƒ ƒ,Ž,ƒƒƒƒ,Ž,

lplpvAudioPtr2
ƒƒƒ,,,ƒƒƒƒ ƒƒƒƒ,2”–,—ˆ,,ƒƒƒƒ ,,ƒƒƒ ƒ,’,NULL,
*lplpvAudioPtr1*ƒƒƒ ƒ,ƒƒƒƒ ƒƒƒƒ,ƒƒƒ,,,—ˆ‘‘,Ž,

lpdwAudioBytes2
lplpvAudioPtr2 ƒƒƒ ƒ,ƒƒƒƒ,,,ƒƒƒ ,Ž,• ,ƒƒƒƒ
lplpvAudioPtr2,NULL, ,,’,0,,,

dwFlags
ƒƒƒ,ƒƒƒƒ, ,,ƒƒƒ ˆ‰,’,Ž,,,

DSBLOCK_FROMWRITECURSOR

Œ , ,,,ƒ ƒƒ,,ƒƒƒ, ,
,,,,,**IDirectSoundBuffer::GetCurrentPosition**,Œ, ,,••—
,,, ,,ƒƒƒ,Ž’,,,,*dwWriteCursor*ƒƒƒ ƒ,–Ž,,, ,,ƒƒƒ,”ˆ,Ž’,,,

のののののの *lplpvAudioBytes2* NULL のの

lplpvAudioPtr2 *lpdwAudioBytes2* NULL
DirectSound の

IDirectSoundBuffer::Lock
DirectSound **IDirectSoundBuffer::Unlock**

Warning

,,*ƒƒƒƒ*, ,,,*ƒƒƒƒ*,,,•, ,,*ƒƒƒƒ*,,*ƒƒƒƒƒ ƒ*,“, ,, ,,, ,,,DirectSoundBuffer*ƒƒƒƒƒ ƒ*,—Œ,*ƒƒƒƒ ƒ ƒ*,Š,,,,, ,,*ƒ ƒ*,–
Œ,,, ,,,, *ƒƒƒƒ*,*ƒƒƒ ƒ ƒƒƒ*,”’,,,,, *ƒƒƒƒ*,*ƒƒƒ ƒƒƒƒƒƒƒ*“,*ƒƒƒƒƒ ƒƒƒƒ*,*ƒƒ ƒƒ*,,, **IDirectSoundBuffer::Unlock**,Œ, ,,,, ,,*ƒƒƒƒƒ ƒƒƒƒ*,*ƒƒƒ ƒ ƒƒƒ*,“‘,, ,

IDirectSoundBuffer, **IDirectSoundBuffer::GetCurrentPosition**, **IDirectSoundBuffer::Unlock**

IDirectSoundBuffer::Play

```
HRESULT Play(DWORD dwReserved1, DWORD dwReserved2,
    DWORD dwFlags);
```

の

- Œ,,,DS_OK Ž”,,,Ž,*ƒƒ ƒ ƒ*,•,

 DSERR_BUFFERLOST

 DSERR_INVALIDCALL

 DSERR_INVALIDPARAM

 DSERR_PRIOLEVELNEEDED

dwReserved1

,,*ƒƒƒ ƒ*,—— ,,,, ,,’,0,,,,,,,,,

dwReserved2

,,*ƒƒƒ ƒ*,—— ,,,, ,,’,0,,,,,,,,,

dwFlags

ƒƒƒƒ, •–,Ž’,, ˆ‰,’,Ž,,,

DSBPLAY_LOOPING

ƒ ƒƒƒ ƒƒƒƒ, —ˆ’,,“’,,, *ƒƒƒƒ*,ŠŽˆ’,, , , , ’Ž,–
Ž,,,,Œ,•, ,,*ƒƒƒ*,*ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ*, ’, ’,,•—,,,

の □□□□□□□□□□

DSBPLAY_LOOPING の

のの

Note ,,ƒƒƒƒ ƒƒƒƒ,,,,, ƒƒƒƒ ƒƒƒ,,,ƒƒƒƒ,Œ, ,,,‘,**IDirectSound::SetCooperativeLevel**ƒƒƒƒ,Œ, , ‹’ƒƒƒ,Ž’,,•—
,,, ’ ,DSSCL_NORMAL,Ž’,, ,,**IDirectSound::SetCooperativeLevel**ƒƒƒƒ,
Œ, ,,,,,,,
IDirectSoundBuffer::Playƒƒƒƒ,**DSERR_PRIOLEVELNEEDED**ƒƒ ’,•,

IDirectSoundBuffer, **IDirectSound::SetCooperativeLevel**

IDirectSoundBuffer::Restore

```
HRESULT Restore();
```

の DirectSoundBuffer

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

DSERR_BUFFERLOST

DSERR_INVALIDCALL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

DirectSoundBuffer::Restore

DSSCL_WRITEPRIMARY DSSCL_WRITEPRIMARY

DirectSound の

IDirectSoundBuffer::Lock **IDirectSoundBuffer::Play**
の
の DSERR_BUFFERLOST
IDirectSoundBuffer::GetStatus

IDirectSoundBuffer, **IDirectSoundBuffer::Lock**, **IDirectSoundBuffer::Play**, **IDirectSoundBuffer::GetStatus**

IDirectSoundBuffer::SetCurrentPosition

```
HRESULT SetCurrentPosition(DWORD dwNewPosition);
```

のの の

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

DSERR_INVALIDCALL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

dwNewPosition
ƒƒƒƒ ƒƒƒƒ, ,Ž—,,,ƒƒƒ’ˆ,ƒƒƒƒƒ’

の

ののの
IDirectSoundBuffer::Play

IDirectSoundBuffer, **IDirectSoundBuffer::GetCurrentPosition**, **IDirectSoundBuffer::Play**

IDirectSoundBuffer::SetFormat

```
HRESULT SetFormat(LPWAVEFORMATEX lpfxFormat);
```

ののの

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_BADFORMAT

  DSERR_INVALIDCALL

  DSERR_INVALIDPARAM

  DSERR_OUTOFMEMORY

  DSERR_PRIOLEVELNEEDED

  DSERR_UNSUPPORTED

lpfxFormat
WAVEFORMATEX ‘‘,,ƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ, ,,ŒŽ,•,

IDirectSoundBuffer::SetFormat
ののDirectSound

PCM のの DSSCL_NORMAL

の DirectSoundBuffer

DirectSound PCM

IDirectSoundBuffer, **IDirectSoundBuffer::GetFormat**

IDirectSoundBuffer::SetFrequency

```
HRESULT SetFrequency(DWORD dwFrequency);
```

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_CONTROLUNAVAIL

  DSERR_GENERIC

  DSERR_INVALIDPARAM

  DSERR_PRIOLEVELNEEDED

dwFrequency
ƒƒƒƒƒƒ ƒ ƒƒƒ, ,,,,,ˆ’,Ž” ’ˆ,ƒƒƒ(Hz) ’,”ˆ,100,, 100,000,,,,,

’,0, Ž” ,Œ ,ƒƒƒƒŒŽ,ƒƒƒƒ,,, ,,ŒŽ,**IDirectSound::CreateSoundBuffer**ƒƒƒƒ,Ž’,,,,,

のののの

IDirectSoundBuffer, **IDirectSound::CreateSoundBuffer**, **IDirectSoundBuffer::GetFrequency**, **IDirectSoundBuffer::Play**, **IDirectSoundBuffer::SetFormat**

IDirectSoundBuffer::SetPan

```
HRESULT SetPan(LONG lPan);
```

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_CONTROLUNAVAIL

  DSERR_GENERIC

  DSERR_INVALIDPARAM

  DSERR_PRIOLEVELNEEDED

lPan

‰,ƒƒƒƒƒ,‘‘ƒƒƒ ƒ,Ž“,, ,,’,”ˆ,-10,000,, 10,000,,, 100dB,”ˆ,Ž’,,

lPan の 0 の(0dB)の-2173 の 21.73 dB
870 の 8.7 dB

-10,000 のの10,000 のの

IDirectSoundBuffer, **IDirectSoundBuffer::GetPan**, **IDirectSoundBuffer::GetVolume**, **IDirectSoundBuffer::SetVolume**

IDirectSoundBuffer::SetVolume

```
HRESULT SetVolume(LONG lVolume);
```

の

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_CONTROLUNAVAIL

  DSERR_GENERIC

  DSERR_INVALIDPARAM

  DSERR_PRIOLEVELNEEDED

lVolume

ƒƒƒƒ ƒƒƒƒ,—‹,,, ‹,ƒƒƒ ƒ ’,”ˆ,0(0dB(dB) ƒƒƒ ƒ• ,,)
,,-10,000 -100Db Ž ,–‰ ,,,,, DirectSound,Œ ,‘•,ƒƒ ƒ,,,,,

の 100dB 0 のの

のののの10dB の 20dB 4 の 1
の DirectSound

IDirectSoundBuffer, **IDirectSoundBuffer::GetPan**, **IDirectSoundBuffer::GetVolume**, **IDirectSoundBuffer::SetPan**

IDirectSoundBuffer::Stop

```
HRESULT Stop();
```

の

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_INVALIDPARAM

  DSERR_PRIOLEVELNEEDED

IDirectSoundBuffer::Stop ののの
IDirectSoundBuffer::Play

IDirectSoundBuffer::Stop

IDirectSoundBuffer::Play の
の
の
dB

IDirectSoundBuffer, **IDirectSoundBuffer::Play**

IDirectSoundBuffer::Unlock

```
HRESULT Unlock(LPVOID lpvAudioPtr1, DWORD dwAudioBytes1,
    LPVOID lpvAudioPtr2, DWORD dwAudioBytes2);
```

の

- Œ,,,DS_OK Ž”,,,Ž,ƒƒ ƒ ƒ,•,

  DSERR_INVALIDCALL

  DSERR_INVALIDPARAM

  DSERR_PRIOLEVELNEEDED

lpvAudioPtr1
IDirectSoundBuffer::Lockƒƒƒƒ, *lplpvAudioPtr1*ƒƒƒ ƒ,Ž“,,,• ,,ƒƒƒƒ

dwAudioBytes1
Ž ,*lpvAudioPtr1*ƒƒƒ ƒ, ,,,,,ƒƒƒ **IDirectSoundBuffer::Lock**ƒƒƒƒ,,,,•,, ,ƒƒƒ ,‰,,,,,,,,

lpvAudioPtr2
IDirectSoundBuffer::Lockƒƒƒƒ, *lplpvAudioPtr2* ƒƒƒ ƒ,Ž“,,,• ,,ƒƒƒƒ

dwAudioBytes2
Ž ,*lpvAudioPtr2*
ƒƒƒ ƒ, ,,,,,ƒƒƒ **IDirectSoundBuffer::Lock**ƒƒƒƒ,,,,•,,,ƒƒƒ ,‰
,,,,,,,,

IDirectSoundBuffer::Lock
lpvAudioPtr1 lpvAudioPtr2 **IDirectSoundBuffer::Unlock**

dwAudioBytes1 dwAudioBytes2 のの

IDirectSoundBuffer, **IDirectSoundBuffer::GetCurrentPosition**, **IDirectSoundBuffer::Lock**

□**“**

DS3DBUFFER

```
typedef struct {
    DWORD     dwSize;
    D3DVECTOR  vPosition;
    D3DVECTOR  vVelocity;
    DWORD     dwInsideConeAngle;
    DWORD     dwOutsideConeAngle;
    D3DVECTOR  vConeOrientation;
    LONG      lConeOutsideVolume;
    D3DVALUE     flMinDistance;
    D3DVALUE     flMaxDistance;
    DWORD     dwMode;
} DS3DBUFFER;
```

のののの
IDirectSound3DBuffer::GetAllParameters
IDirectSound3DBuffer::SetAllParameters

dwSize
,, ‘‘,ƒƒƒ’ˆ,ƒƒƒ

vPosition
,,ƒƒƒƒ ƒƒƒƒ,Œ ,ˆ’,‹ ,, **D3DVECTOR** ‘‘

vVelocity
,,ƒƒƒƒ ƒƒƒƒ,Œ ,‘“,‹ ,, **D3DVECTOR** ‘‘

dwInsideConeAngle

"'*ƒƒƒƒ ƒƒƒƒƒƒƒƒ ƒ ƒ*,Š"

dwOutsideConeAngle

Š'*ƒƒƒƒ ƒƒƒƒƒƒƒƒ ƒ ƒ*,Š"

vConeOrientation

,,,,*ƒƒƒƒƒƒƒƒ ƒƒƒƒƒƒƒƒ ƒ ƒ*,Œ ,•Œ,‹ ,, **D3DVECTOR** ''

lConeOutsideVolume

Š•*ƒƒƒ ƒ,ƒ ƒ*

flMinDistance

‹—

flMaxDistance

'‹—

dwMode

,,*ƒƒƒƒ —ƒ ƒ* Ž,',Ž,,,

DS3DMODE_DISABLE

3D*ƒƒƒƒ* —,–Œ,,, *ƒƒƒƒ,ƒƒƒ* ,",' ,,•,,,,,,Š,,

DS3DMODE_HEADRELATIVE

ƒƒƒƒƒƒƒ ƒ(ˆ' '" •Œ)
,*ƒƒƒ* ,*ƒƒƒ ƒ*,''ŠŒ,,, ,,*ƒ ƒ*,, *ƒƒƒƒ*, '*ƒƒƒ ƒ,ƒƒƒ ƒƒƒ ƒ*,• ,,,,,,
Ž"", ,,,,, ''*ƒƒƒ ƒ*,•,,,,,,,,,

DS3DMODE_NORMAL

• , — ,,,*ƒƒƒƒƒƒ ƒ*,,,,,,

DS3DLISTENER

```
typedef struct {
    DWORD       dwSize;
    D3DVECTOR   vPosition;
    D3DVECTOR   vVelocity;
    D3DVECTOR   vOrientFront;
    D3DVECTOR   vOrientTop;
    D3DVALUE    flDistanceFactor;
    D3DVALUE    flRolloffFactor;
    D3DVALUE    flDopplerFactor;
} DS3DLISTENER;
```

のののの

IDirectSound3DListener::GetAllParameters
IDirectSound3DListener::SetAllParameters

dwSize

,, '',*ƒƒƒ*'ˆ,*ƒƒƒ*

vPosition, **vVelocity**, **vOrientFront**, **vOrientTop**

ƒƒƒ ,ˆ' '" '•Œ,,, •Œ,,,,,‹ ,, **D3DVECTOR** ''

flDistanceFactor, **flRolloffFactor**, **flDopplerFactor**

,,,,Œ ,‹— *ƒ ƒƒƒ*,,,*ƒƒƒƒ* —'

DSBCAPS

```
typedef struct _DSBCAPS {
    DWORD dwSize;
    DWORD dwFlags;
    DWORD dwBufferBytes;
    DWORD dwUnlockTransferRate;
    DWORD dwPlayCpuOverhead;
} DSBCAPS, *LPDSBCAPS;
```

IDirectSoundBuffer::GetCaps DirectSound
の

dwSize

,, ‘‘,ƒƒƒ’ˆ,ƒƒƒ

dwFlags

ƒƒƒƒ ƒƒƒƒƒƒ”—,Ž’,,ƒƒƒ

DSBCAPS_CTRL3D

ƒƒƒƒ,,, Œ,Ž—,,ƒƒƒƒƒ ƒƒƒƒ,,,

DSBCAPS_CTRLFREQUENCY

ƒƒƒƒ,Ž” Œ”—,Ž,,,,,,,,,

DSBCAPS_CTRLPAN

ƒƒƒƒ,ƒƒ Œ”—,Ž,,,,,,,,,

DSBCAPS_CTRLVOLUME

ƒƒƒƒ,ƒƒƒ ƒ Œ”—,Ž,,,,,,,,,

DSBCAPS_GETCURRENTPOSITION2

IDirectSoundBuffer::GetCurrentPosition , ƒ ƒƒ, ,,“ ,Ž—
,,,,,,,,,,Ž, DirectX ,,DirectSound
,, ,,,ƒ ƒƒ,’Œ,,,, ƒ ƒƒ,ƒƒƒƒ ƒ ƒ,ƒƒƒƒ ƒ,,‰, Ž ,‰,,, —
,, ,,,, Œ ,, DSBCAPS_GETCURRENTPOSITION2ƒƒƒ,Ž’,,,, ƒƒƒƒ
ƒƒƒ,,,,, Š, ˆ’,Ž“,‰”,,,,
,,ƒƒƒ,Ž’,,,, ‘,,,,‰‹,“ ,ŒŠ ,,,,ˆŽ,,,,, ,,ƒƒƒ,‰‹,—
,,,,ƒƒƒƒ ƒ ƒ ƒƒƒƒ ƒ,,,,,,,,,’ˆ,,,, ,, DirectSound
ƒƒƒƒ,,,, DirectX,,,,,ƒ ƒƒƒ,DirectSound,, ƒ ƒƒ, Š,,,

DSBCAPS_GLOBALFOCUS

ƒƒƒƒ,ƒƒ ƒƒ,ƒƒƒƒ ƒƒƒƒ,,,
,,ƒƒƒ,ƒƒƒ,,,,,,, ,,ƒ ƒ,‘,ƒƒƒƒ ƒƒƒ,ƒƒ ƒƒ, ,‘,,, ,,ƒƒƒƒ ƒƒƒ,DirectSound ,Ž, DirectSound ,Ž,,,,ƒƒƒƒ ƒƒƒ,,,ƒƒƒƒ, ,‘,,,,,,,,
,,, ƒƒ ƒƒ, DSSCL_EXCLUSIVE
,,,, DSSCL_WRITEPRIMARYƒƒƒ,‹’ƒƒƒ,Ž,DirectSound
ƒƒƒƒ ƒƒƒ, ,‘,, ,—Š,,, ,, ‘,ƒƒƒƒ ƒƒƒ,,,ƒƒ ƒƒ,‰,•,,,,,,

DSBCAPS_LOCHARDWARE

DSBCAPS_STATIC ,Ž’,,,,,, ƒƒƒƒ,‹ “,ƒ ƒƒƒƒ ƒƒƒƒƒ,Ž—,,
,,ƒƒƒƒ,ƒ ƒƒƒƒ ƒƒƒƒƒ,ƒƒ ƒ,,,,,, ,,,,•—,,,,ƒ ƒƒƒƒ ƒƒƒ,——
•‰”, , **IDirectSound::CreateSoundBuffer** ,Œ, ,,Ž”,,
ƒƒƒƒ ƒƒƒ,ƒƒƒƒƒ ƒƒƒƒ,,,ƒƒƒƒ,,,,——‰”,,,,,,• ,,•—,,,

DSBCAPS_LOCSOFTWARE

DSBCAPS_STATIC ,Ž’,,ƒ ƒƒƒƒ ƒƒ ƒ,——

‰”,,,,, ƒƒƒƒ,‹ “,ƒƒƒƒƒƒ ƒƒƒ,Š”,, ƒƒƒƒƒƒ ƒƒƒƒƒ,Ž—,,

DSBCAPS_PRIMARYBUFFER

ƒƒƒƒ,ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ,,,,,,•,
,,,,’,Ž’,,,,,, *ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ*, ,,,

DSBCAPS_STATIC

ƒƒƒƒ,ƒƒƒƒƒƒ ƒƒƒƒ ƒ ƒ,,,,Ž,,,,,,•,
ˆ”,,,,,*ƒƒƒƒ*,,“*ƒ ƒ*,,, ,,,‰“, ,,, ,,,,*ƒƒƒƒ,ƒ ƒƒƒƒ ƒƒƒ*,—‹,,

DSBCAPS_STICKYFOCUS

ƒƒƒƒ ƒƒƒƒ,ƒƒ ƒƒ,“ ,•,,
,,*ƒƒƒ*,**IDirectSound::CreateSoundBuffer**,Œ, ,,Ž’,,,,,,,,
,,*ƒƒƒ,ƒƒƒ*,,,, ,,*ƒ ƒ*,‘,DirectSound ,Ž,,,,,*ƒƒƒƒ ƒƒƒ*, ,‘,,,
DirectSound ,Ž,,,,*ƒƒƒƒ ƒƒƒ*,,*ƒƒƒƒƒ ƒƒ ƒƒ ƒƒƒƒ*, ,‘,,,,,,,,
,, ‘,*ƒƒƒƒ ƒƒƒ*,• *ƒƒƒƒ*, ‰,,,, *ƒƒƒƒƒ ƒƒ ƒƒ ƒƒƒƒ*,–
,‘,, ,,,,*ƒ ƒ,ƒ ƒ*
,,,,*ƒƒƒƒ,ƒƒƒ*,,Š,*ƒƒƒƒƒƒƒƒ*,•,,,,Ž *ƒ ƒ*ˆŠ,*ƒƒƒƒ ƒƒƒ* ,,,,‰‰
(ActiveMovie™),,,——, ,,, *ƒ ƒ*,,,‘, DirectSound
ƒƒƒƒ ƒƒƒ, ,‘,,,, • ,*ƒƒƒƒƒ ƒƒ ƒƒ*—•,*ƒƒƒƒ ƒƒƒƒ*,,,, ‰,,,

dwBufferBytes

,,*ƒƒƒƒ,ƒƒƒ*’ˆ,*ƒƒƒ*

dwUnlockTransferRate

IDirectSoundBuffer::Unlock,Œ, ,,,,,,*ƒƒƒƒ ƒƒƒ,ƒ ƒ*,“‘,,,*ƒ ƒ*, ,•–
,k*ƒƒƒ*,Ž’,, *ƒƒƒƒƒ ƒƒƒ,ƒƒƒƒ ƒƒƒ*, **IDirectSoundBuffer::Unlock**,Ž ,,,,,
—,,ŽŠ,Œ’,,,, ,,’,Ž,,,,,,, *ƒƒƒƒƒƒƒ*,”’,,,,,*ƒƒƒƒƒƒ ƒƒƒƒ*,,,,*ƒ ƒ*, —,•
—,,,” , ,
ƒ ƒƒƒƒ ƒƒƒƒ,,,,*ƒ ƒ*, *ƒƒƒƒ,ƒƒƒƒ ƒ ƒ,ƒ ƒ*,,,‰” ,,,,, ,,’,,,‰” ,,, ,
,,““““,Œ’,,, ,,,

dwPlayCpuOverhead

,,*ƒƒƒƒ ƒƒƒƒ,ƒƒƒƒƒ*,,,,•—,*ƒ ƒ ƒƒƒ*,Ž, —*ƒƒƒƒ,ƒ ƒƒƒ ƒ*,Ž’,,
ƒ ƒƒƒƒ ƒƒƒƒ,,,,, ,,*ƒƒƒ,ƒƒƒƒƒ,ƒƒƒƒ ƒƒƒƒ*,,,, ,,,,, ,,,,
ƒƒƒƒƒƒ ƒƒƒƒ,,,,, ,,*ƒƒƒ,ƒƒƒƒ ƒƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒƒ*,‘“,ˆ‘,,,

DSBCAPS **IDirectSound::CreateSoundBuffer**
DSBUFFERDESC
の

DSBCAPS **dwFlags** **DSBUFFERDESC**
の **DSBCAPS**
DSBCAPS_LOCHARDWARE DSBCAPS_LOCSOFTWARE
の **DSBUFFERDESC**
ののの

IDirectSound::CreateSoundBuffer, **IDirectSoundBuffer::GetCaps**

DSBUFFERDESC

```
typedef struct _DSBUFFERDESC{
    DWORD           dwSize;
    DWORD           dwFlags;
    DWORD           dwBufferBytes;
    DWORD           dwReserved;
    LPWAVEFORMATEX  lpwfxFormat;
} DSBUFFERDESC, *LPDSBUFFERDESC;
```

DirectSoundBuffer
IDirectSound::CreateSoundBuffer

dwSize

,, ‘‘,*fff*’ˆ,*fff*

dwFlags

,, DirectSoundBuffer *ffffff*, ,,,,Š,”—,Ž•,, ˆ‰,,,,,,,,• ,Ž’,,,,

DSBCAPS_CTRL3D

ffff,,, Œ,Ž,*fffff* *ffff*,,,

DSBCAPS_CTRLALL

ffff,,,,, Œ”—,Ž,,,,,,,,,

DSBCAPS_CTRLDEFAULT

ffff,*fffff*,*ffff* *f* *fffff*,Ž,,,,,, ,,,DSBCAPS_CTRLPAN DSBCAPS_CTRLVOLUME,,,DSBCAPS_CTRLFREQUENCY*fff*,Ž’,, ,,,“,,,,

DSBCAPS_CTRLFREQUENCY

ffff,Ž” Œ”—,Ž,,,,,,,,,

DSBCAPS_CTRLPAN

ffff,*ff* Œ”—,Ž,,,,,,,,,

DSBCAPS_CTRLVOLUME

ffff,*fff* *f* Œ”—,Ž,,,,,,,,,

DSBCAPS_GETCURRENTPOSITION2

IdirectSoundBuffer::GetCurrentPosition , *f* *ff*, ,, ’,Ž— ,,,,,,,,,,Ž, DirectX ,,DirectSound ,, ,,,*f* *ff*,’Œ,,,, *f* *ff*,*ffff* *f* *f*,*ffff* *f*,,‰, Ž ,‰,,, — ,, ,,,, Œ ,, DSBCAPS_GETCURRENTPOSITION2*fff*,Ž’,, *ffff* *fff*,,,,, Š, ˆ’,Ž“,‰”,,,, ,,*fff*,Ž’,,,, ‘,,,,‰‹,“ ,ŒŠ ,,,,ˆŽ,,,,, ,,*fff*,‰‹,— ,,,,*ffff* *f* *f* *ffff* *f*,,,,,,,,,’ˆ,,,, ,,DirectSound *ffff*,,,, DirectX,,,,,*f* *fff*,DirectSound,, *f* *ff*, Š,,,

DSBCAPS_GLOBALFOCUS

ffff,*ff* *ff*,*ffff* *ffff*,,, ,,*fff*,*fff*,,,,,,, ,,*f* *f*,‘,*ffff* *fff*,*ff* *ff*, ,‘,,, ,,*ffff* *fff*,DirectSound ,Ž, DirectSound ,Ž,,,,*ffff* *fff*,,,*ffff*, ,‘,,,,,,,, ,,, *ff* *ff*, DSSCL_EXCLUSIVE ,,,, DSSCL_WRITEPRIMARY*fff*,‹’*fff*,Ž,DirectSound *ffff* *fff*, ,‘,, ,—Š,,, ,, ‘,*ffff* *fff*,,,*ff* *ff*,‰,•,,,,,,

DSBCAPS_LOCHARDWARE

DSBCAPS_STATIC ,Ž’,,,,,, *ffff*,‹ “,*f* *ffff* *fffff*,Ž—,, ,,*ffff*,*f* *ffff* *fffff*,*ff* *f*,,,,,, ,,,,•—,,,,*f* *ffff* *fff*,—— •‰”, , **IDirectSound::CreateSoundBuffer** ,Œ, ,,Ž”,, *ffff* *fff*,*fffff* *ffff*,,,*ffff*,,,,——‰”,,,,,,• ,,•—,,,

DSBCAPS_LOCSOFTWARE

DSBCAPS_STATIC ,Ž’,,*f ffff ff f*,——
‰”,,,,, *ffff*,‹ “,*ffffff fff*,Š”,, *ffffff fffff*,Ž—,,

DSBCAPS_PRIMARYBUFFER

ffff,*fffff ffff ffff*,,,,,,•,
,,,,’,Ž’,,,,,, *fffff ffff ffff*, ,,,

DSBCAPS_STATIC

ffff,*ffffff ffff f f*,,,,Ž,,,,,,•,
ˆ”,,,,,*ffff*,,“*f f*,,, ,,,‰“, ,,, ,,,,*ffff*,*f ffff fff*,—‹,,

DSBCAPS_STICKYFOCUS

ffff ffff,*ff ff*,“ ,•,,
,,*fff*,**IDirectSound::CreateSoundBuffer**,Œ, ,,Ž’,,,,,,,,
,,*fff*,*fff*,,,, ,,*f f*,‘,DirectSound ,Ž,,,,,*ffff fff*, ,‘,,,
DirectSound ,Ž,,,,*ffff fff*,,,*fffff ff ff ffff*, ,‘,,,,,,,,
,, ‘,*ffff fff*,• *ffff*, ‰,,,, *fffff ff ff ffff*,–
,‘,, ,,,,*f f*,*f f*
,,,,*ffff*,*fff*,,Š,*ffffffff*,•,,,,Ž *f f*ˆŠ,*ffff fff* ,,,,‰‰
(ActiveMovie™),,,——, ,,, *f f*,,,‘, DirectSound
ffff fff, ,‘,,,, • ,*fffff ff ff*—•,*ffff ffff*,,,, ‰,,,

dwBufferBytes

,,*ffff*,*fff*’ˆ,*fff* ,,’,*fffff ffff*, ,,,,,,,,,,,,,,

dwReserved

,,*fff*,——,,,,, Ž—,,,,

lpwfxFormat

ffff,,,,wave*ff fff*,Ž’,,,, ‘‘,,*ffff* ,,’,*fffff ffff*,,NULL,,,,,,,,,
ffff fff,*fffff ffff*,*ff fff*,*fff*,,,,, **IDirectSoundBuffer::SetFormat**
,Ž,,,,,,,

dwFlags DSBCAPS_LOCHARDWARE
DSBCAPS_LOCSOFTWARE
DSBCAPS_LOCHARDWARE の
DSBCAPS_LOCSOFTWARE

の **DSBCAPS** **dwFlags**
DirectSoundBuffer

dwBufferBytes
DirectSound のの
のの

IDirectSound::CreateSoundBuffer

DSCAPS

```
typedef struct _DSCAPS {
    DWORD  dwSize;
    DWORD  dwFlags;
    DWORD  dwMinSecondarySampleRate;
    DWORD  dwMaxSecondarySampleRate;
    DWORD  dwPrimaryBuffers;
    DWORD  dwMaxHwMixingAllBuffers;
    DWORD  dwMaxHwMixingStaticBuffers;
    DWORD  dwMaxHwMixingStreamingBuffers;
    DWORD  dwFreeHwMixingAllBuffers;
    DWORD  dwFreeHwMixingStaticBuffers;
    DWORD  dwFreeHwMixingStreamingBuffers;
    DWORD  dwMaxHw3DAllBuffers;
    DWORD  dwMaxHw3DStaticBuffers;
    DWORD  dwMaxHw3DStreamingBuffers;
    DWORD  dwFreeHw3DAllBuffers;
    DWORD  dwFreeHw3DStaticBuffers;
    DWORD  dwFreeHw3DStreamingBuffers;
    DWORD  dwTotalHwMemBytes;
    DWORD  dwFreeHwMemBytes;
    DWORD  dwMaxContigFreeHwMemBytes;
    DWORD  dwUnlockTransferRateHwBuffers;
    DWORD  dwPlayCpuOverheadSwBuffers;
    DWORD  dwReserved1;
    DWORD  dwReserved2;
} DSCAPS, *LPDSCAPS;
```

IDirectSound::GetCaps DirectSound の

dwSize

,, ‘‘,ƒƒƒ’ˆ,ƒƒƒ

dwFlags

ƒƒƒƒ,”—,Ž’,, ˆ‰,,,,,,,,• ,Ž’,,,

DSCAPS_CERTIFIED

,,ƒƒƒƒ,Microsoft,,,,ŽŒ,,• ,,,,,

DSCAPS_CONTINUOUSRATE

ƒƒƒƒ,**dwMinSecondarySampleRate** ,**dwMaxSecondarySampleRate**
ƒƒƒ’,Š,,,,,ƒƒƒƒƒƒƒƒ ƒ,ƒƒ ƒ,, ˆ”,,,,Ž , —ƒ ƒ,—‹,,Ž” ,
,,ƒƒƒ Hz ,’,,,,,,ˆ–,,

DSCAPS_EMULDRIVER

ƒƒƒƒ,,ƒƒƒƒ ƒ,,,DirectSound
ƒƒƒƒ,,, ,,,ƒƒ ƒƒƒ ƒ ƒ ƒƒƒ‹”,’,,ƒƒƒƒ ƒ,,,,, Ž ”—,’‰,—‘,,,

DSCAPS_PRIMARY16BIT

ƒƒƒƒ,,,ƒƒƒƒƒƒƒƒƒ,ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ,ƒƒ ƒ,,

DSCAPS_PRIMARY8BIT

ƒƒƒƒ,,ƒƒƒƒƒƒƒƒƒ,ƒƒƒƒƒ ƒƒƒƒ,ƒƒ ƒ,,

DSCAPS_PRIMARYMONO

ƒƒƒƒ,’‰ƒƒƒƒƒ ƒƒƒƒ,ƒƒ ƒ ,,

DSCAPS_PRIMARYSTEREO

ƒƒƒƒ,ƒƒƒƒƒƒƒƒƒ ƒƒƒƒ,ƒƒ ƒ,,

DSCAPS_SECONDARY16BIT

ƒƒƒƒ,16ƒƒƒƒƒƒƒƒƒ,ƒ ƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒ,ƒƒ ƒ,,

DSCAPS_SECONDARY8BIT

ƒƒƒƒ‚‚ƒƒƒƒƒƒƒ‚ƒ ƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒ‚ƒƒ ƒ‚‚

DSCAPS_SECONDARYMONO

ƒƒƒƒ‚ƒ ƒƒƒƒ ƒƒƒƒƒ’‰ƒƒƒƒƒ ƒƒƒƒ‚ƒƒ ƒ‚‚

DSCAPS_SECONDARYSTEREO

ƒƒƒƒ‚ƒ ƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒ‚ƒƒ ƒ‚‚

dwMinSecondarySampleRate **dwMaxSecondarySampleRate**
‚‚ƒƒƒƒ‚ƒ ƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒ‚ƒƒ ƒ‚‚‚ ’ ‚ƒƒƒƒƒƒƒ ƒ‚Ž’

dwPrimaryBuffers
ƒƒ ƒ‚‚‚ƒƒƒƒƒ ƒƒƒƒ‚ ‰‚ƒƒ ƒ‚‚‚‚‚‚

dwMaxHwMixingAllBuffers
ƒ ƒƒƒƒ‚ƒƒƒƒƒ‚‚‚ƒƒƒƒ‚ Œ‚ ‚Ž’‚‚ ‚‚ƒƒƒ‚ **dwMaxHwMixingStaticBuffers** ‚ **dwMaxHwMixingStreamingBuffers** ‚ Œ‚‚ ‚‚
ƒƒ ƒ‚ƒƒ ƒƒƒ‚‚‚‚” ‚‚

dwMaxHwMixingStaticBuffers
ƒƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ‚ ‘ ‚Ž’‚‚

dwMaxHwMixingStreamingBuffers
ƒƒƒ ƒ ƒƒƒƒ ƒƒƒƒ‚ ‘ ‚Ž’‚‚

dwFreeHwMixingAllBuffers, **dwFreeHwMixingStaticBuffers**, **dwFreeHwMixingStreamingBuffers**
Š‚“‚‚‚‚‚‚ƒƒƒƒ‚ ƒ ƒƒƒƒ ƒƒƒƒƒ”—‚ ’
ƒƒƒƒ ƒƒƒ‚ ƒ ƒƒƒƒ ƒƒ ƒ‚ƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ‚Š‚“‚‚‚‚——
‰”‚‚‚‚‚‚”’‚‚‚‚ ‚‚‚’‚Ž‚‚‚‚‚‚ “‚‚‚‚‚’‚ ‘‚ƒƒƒƒƒ”—
‚Ž’‚‚ƒƒƒ‚”Š‚‚‚ ‚‚‚Š‚“‚‚‚‚‚ƒƒ ƒ‚“’‚‚‚‚‚‚‚

dwMaxHw3DAllBuffers, **dwMaxHw3DStaticBuffers**, **dwMaxHw3DStreamingBuffers**
ƒƒƒƒ‚ƒ ƒƒƒƒ‚‚’ˆ”—‚‹ ‚‚‚‚‚‚ ‚ƒƒ ƒ‚‚‚‚‚‚

dwFreeHw3DAllBuffers, **dwFreeHw3DStaticBuffers**, **dwFreeHw3DStreamingBuffers**
Š‚“‚‚‚‚‚‚ƒƒƒƒ‚ ƒ ƒƒƒƒ‚‚’ˆŒ‚”—‚‹ ‚‚‚‚‚‚ ‚ƒƒ ƒ‚‚‚‚‚‚

dwTotalHwMemBytes
ƒƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ‚Š”‚‚ƒƒƒƒ ƒ ƒ ‚ƒƒƒ——‚ƒƒƒ’ˆ‚ƒƒƒ

dwFreeHwMemBytes
ƒƒƒƒ ƒ ƒ ‚ƒƒ ƒƒƒ‚ƒƒƒ’ˆ‚ƒƒƒ

dwMaxContigFreeHwMemBytes
ƒƒƒƒ ƒ ƒ ‚ƒƒ ƒƒƒ‚ ‚‘‚‚˜‘ƒƒƒƒ‚ƒƒƒ’ˆ‚ƒƒƒ

dwUnlockTransferRateHwBuffers
ƒ ƒƒƒƒƒƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒ ƒƒƒ ƒƒƒƒƒƒƒƒ‚”’‚‚‚‚ ‚ƒ ƒ‚“‘‚‚‚ƒ ƒ‚‚•–‚k
ƒƒƒ‚‹ ‚‚ ‚‚ƒ ƒ‚ƒƒƒ‚‚“‘—‚
IDirectSoundBuffer::UnlockƒƒƒƒŒ‚ ‚‚Œ‘ŽŠ‚Œ’‚‚

dwPlayCpuOverheadSwBuffers
ƒƒƒƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒƒƒƒ‚”’‚‚‚ ‚ƒƒƒƒƒ‚‚‚‚•—‚ CPU‚ƒ ƒƒƒ ƒ‚ —
ƒ ƒ ƒƒƒ‚’ ‚‚‚ƒƒƒƒƒ ƒƒƒƒƒƒƒƒ‚‚‚ƒƒƒƒ‘“‚ˆ‘‚‚•‰‚‚

ƒƒƒ,,,,,,ƒƒƒƒƒƒ ƒƒƒƒ,"'ƒ ƒ, ƒ ƒ,,,,,"',,•—,,,,,,,,, "—
,ƒ ƒƒƒƒƒƒƒƒ, ƒ ƒ ƒƒƒ, ƒƒƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,,, —,,,,,,,,,

dwReserved1 **dwReserved2**

,,,,ƒƒƒ,——,,,,, Ž—,,,,

IDirectSound::GetCaps

•'

のの IDirectSound IDirectSoundBuffer
ののの

DS_OK

—‹, Œ,Š—,,

DSERR_ALLOCATED

—‹,— ƒƒƒ,,,,ƒƒ ƒ,',Œ, ,,,,,,,,Ž—',,,,,Ž",,

DSERR_ALREADYINITIALIZED

ƒƒƒƒƒƒ,,,, Š‰,,,,,

DSERR_BADFORMAT

Ž',,,wave,ƒƒ ƒ,,,,

DSERR_BUFFERLOST

ƒƒƒƒƒƒƒ,Ž,,,,, ƒƒƒƒ,,,,,,,,,

DSERR_CONTROLUNAVAIL

Œ, ,,,,,—‹,, Œ ƒƒƒ ƒ ƒƒ,, ,——‰",,,,

DSERR_GENERIC

–Š',ƒƒ , DirectSound ƒƒƒƒƒƒ,',‹,,,

DSERR_INVALIDCALL

,,Š ,,,ƒƒƒƒƒƒ,Œ , ',,,,—Œ,,,,

DSERR_INVALIDPARAM

–Œ,ƒƒƒ ƒ,Š ,",,,

DSERR_NOAGGREGATION

ƒƒƒƒƒƒ,COM ‰,ƒƒ ƒ,,,

DSERR_NODRIVER

Ž—,,ƒƒƒƒ ƒƒƒƒ,——,,,,

DSERR_OTHERAPPHASPRIO

,,',‰‹,,,,,,Ž—,,,,

DSERR_OUTOFMEMORY

DirectSound ƒƒƒƒƒƒ,Œ, ,,—‹,Š—,,,,, •,ƒƒƒ,Š,",,,,,,,,,,,

DSERR_PRIOLEVELNEEDED

Œ, ,,Š , Œ,,,,,•—,,,— ƒƒƒ,Ž,,,,,

DSERR_UNINITIALIZED

',ƒƒƒƒ,Œ, ,,,',**IDirectSound::Initialize**ƒƒƒƒ,Œ, ,,,,,,, ,,,,Œ, ,, Œ,,,

,,

DSERR_UNSUPPORTED

Œ, ,,Š ,,,,,ƒƒ ƒ,,,,,,