

Microsoft[®] DirectX[™] 3

ffffffŠ"fff

„ffffff, < „„„ „, — „„• „„„„„ „ „•, %Ž– –
„„f f, „, < „„„„, Œ, %o<„„„„ „ffffff, „•, ^•, Microsoft Corporation, –„„, <%o
„„„ „<“ <Š“ „„„„, •Ž –„„• „„„„„„
Microsoft, „ffffff, < „„„„ ffff fff, „<„„„ ‚, „< • ‚ Œ „„, ‚“ —
Œ, •Ž„„„ „ffffff, Microsoft, –„„, <%o„„„ „< • ‚ Œ „„, ‚“ —Œ, <%o, —
„„„„„

©1996 Microsoft Corporation. All rights reserved.

Microsoft ActiveMovie Direct3D DirectDraw DirectInput DirectPlay DirectS
ound DirectX MS-DOS Win32 Windows „„, Windows NT, • Microsoft
Corporation, • „„„„, ‚ „„„ „•„„

„„, ‚ •–, %Ž–, ŠŽ, •„„

DirectSound

DirectSound,,,,,.....
 DirectSoundf□fffff.....
 f□fffffŠ—.....
 fffff□fff.....
 fffff□ffff□fff.....
 ffff□ffff.....
 <'fff.....
 ffff,“□.....
 DirectSoundŠ—.....
 DirectSound,“.....
 3Dffff.....
 DirectSoundfff□ffffŠ—.....
 IDirectSoundfff□ffff.....
 IDirectSound3DBufferfff□ffff.....
 IDirectSound3DListenerfff□ffff.....
 IDirectSoundBufferfff□ffff.....
 DirectSound,Ž—.....
 DirectSoundfffff,□□.....
 CoCreateInstance,Ž—,DirectSoundfffff,□□.....
 f□ffff□“,□%.....
 ffff□ffff,□□.....
 ffff ffff,, ,,,.....
 DirectSoundfff ,Ž—.....
 ffff□ffff□,Ž—.....
 ^ wave,Ž—.....
 DirectSoundfffff.....
 Š□.....
 f□ffffŠ□.....
 IDirectSound.....
 IDirectSound3DBuffer.....
 IDirectSound3DListener.....
 IDirectSoundBuffer.....
 □“.....

•

DirectSound,,,,

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

DirectSoundf□fffff

のDirectSound のの DirectXのの

- f fffffŠ—
- fffffff fff
- fffffff□ffff□fff
- ffff□ffff
- ‹fff
- ffff,“□

f□fffffŠ—

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

の

ffffff **ffffff**

の DirectSound DirectSound *IDirectSound*
(COM)ののの

のDirectSound

DirectSound

IDirectSoundBuffer

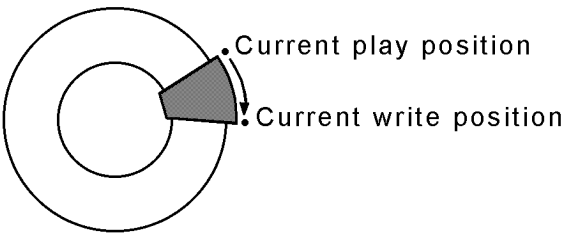
ののの

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

- *DirectSound*ffffff
- *DirectSoundBuffer* fffffff

DirectSoundffffff

の

IDirectSound

DirectSoundCreate

IDirectSound

DirectSound

DirectSound

DirectSoundEnumerate

Windows

の

DirectSound

Note DirectSound, *ffffff*, IDirectSound, *fff*, IDirectSoundBuffer *fff* *fff*
, ' < , , , Cœœ, *fff* ' < , Š , , ,

DirectSoundBuffer *ffffff*

DirectSoundBuffer *IDirectSoundBuffer*
IDirectSound::CreateSoundBuffer DirectSoundBuffer
IDirectSoundBuffer

の DirectSound

IDirectSoundBuffer::Lock

IDirectSoundBuffer::Play

の

DirectSound

IDirectSoundBuffer::Play の

の

の

DirectSoundBuffer
IDirectSound::CreateSoundBuffer DSBCAPS_STATIC
 (の)

PCI の

の

DirectSound

ffffff □ *ffff* □ *fff*

DirectSound

IDirectSound::GetCaps

DirectSound

ffff □ *ffff*

DirectSound

DirectSound HAL

HAL

の

Windows の

Windows の DirectSound HAL

- *f* *fff* < Š , ' , , *ffff* *f* , Ž “ , % •

- f fffŠ, ”%o
- f ffff,—Œ, ,“,‘ ,Ž
- f ffff,—Œ, ,‘ —Š,“

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

Š’ffff
 DirectSound 4
 IDirectSound::SetCooperativeLevel

の
 IDirectSoundBuffer::SetFormat IDirectSound::Compact
 の

DirectSound
 IDirectSoundBuffer::SetFormat IDirectSound::Compact

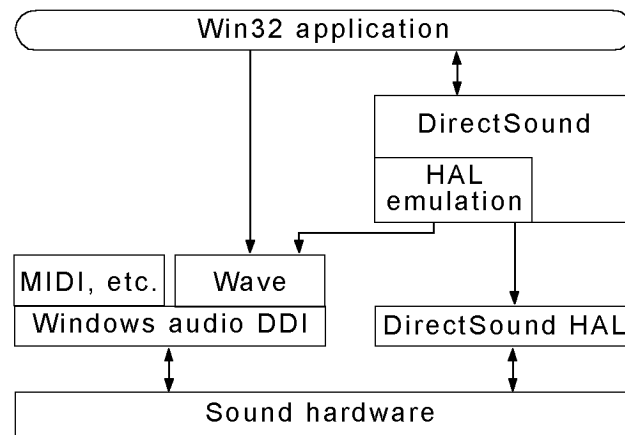
DirectSound DirectSound wave
 wave
 IDirectSoundBuffer::SetFormat ののの (DirectSound
 wave)

の DirectSound
 IDirectSoundBuffer::Lock

(のIDirectSoundBuffer::Restore) の
 の

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

ffff,“□
 の DirectSound ののの



DirectSound HAL

HAL の

HAL の

DirectSound の

DirectSound

DirectSound の

DirectSound
HAL の

DirectSound

の

DirectSoundCreate*IDirectSound**IDirectSoundBuffer***IDirectSound::Release**

の

Win32 SDK の

の

PlaySound**WaveOut**

Note Microsoft Video for Windows, ㊦ avi(audio visual interleaved)ffff,f fff ffff, —,,,ff fff f f fffŠ ,ž— ,,,, ,,,, ffff fff,DirectSound,ž— ,,,,aviffff, ,,,,,, ,,f fff ffff, ,,,, “— , aviffff, ,,,,DirectSoundffffff, ,,,,,, ž,Š ,ff ,*,

㊦ ,,,,,, ffff fff,DirectSoundffffff,%o•,,,,, aviffff, ‘,IDirectSound::Release,㊦, ,, , ffff fff,fff, — ㊦,DirectSoundffffff,,DirectSoundBufferffffff, , , , ,

DirectSoundŠ—

の DirectSound の

- *DirectSound*, “’
- *3Dffff*

DirectSound, “’

の DirectSound ののの

- *fffff*
- *f ffff fffff fff*
- *ffffff ffff,, ,,,ffff*

ffffff

DirectSound の
DirectSound のののDirectSound
の

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

ののののの DirectSound

f fffff fffff fff

DirectSound の

のの

DirectSound

のの

ののののの

ffffff ffff,, ,,,ffff

DirectSound □□□□□□□□ □□□

□□□□□□□□

□□□

□□□

□□□

DirectSound の
(DirectSound

の □□□□□□□□
の□□□ のののののののの

3Dffff

DirectSound

IDirectSound3DBuffer

IDirectSound3DListener

の

のの

3D のの

- %0“”^,”Ž
- fff
- ffff f f
- ’<—, ’<—
- ^,”“
- Direct3D,”“
- “”^,<—‘
- ffff%0Œ,ffff%0Œ

%0“”^,”Ž

のののの

- fff f %0, ,•’,fff ,,—,,, ,,%0, ,,,,,, ,,Œ ,f fff,Œ,,,,,
- “”,, fff ,%0•Œ,”,,,,%0,fff ,%0Ž,”,,, Ž,,,,, , ,,, (,,,Ž•ŽŠ, – 1ff•,,)
- %0 fff ,Œ,,,,,%0, ‘,,,,%0,”,,,,, %0 ,,,, ,,,, ,Ž,•ŒŠ”,ŠŽ,,, ,,, fff ,%0,,%0,,, , Ž,”,%0,”,,,,, %0 ,,,

ののの DirectSound ののの 3D

ののののの

fff□

の 90

90 ののののののDirectSound

IDirectSound3Dlistene

ffff□f□f

□□□□□□□□

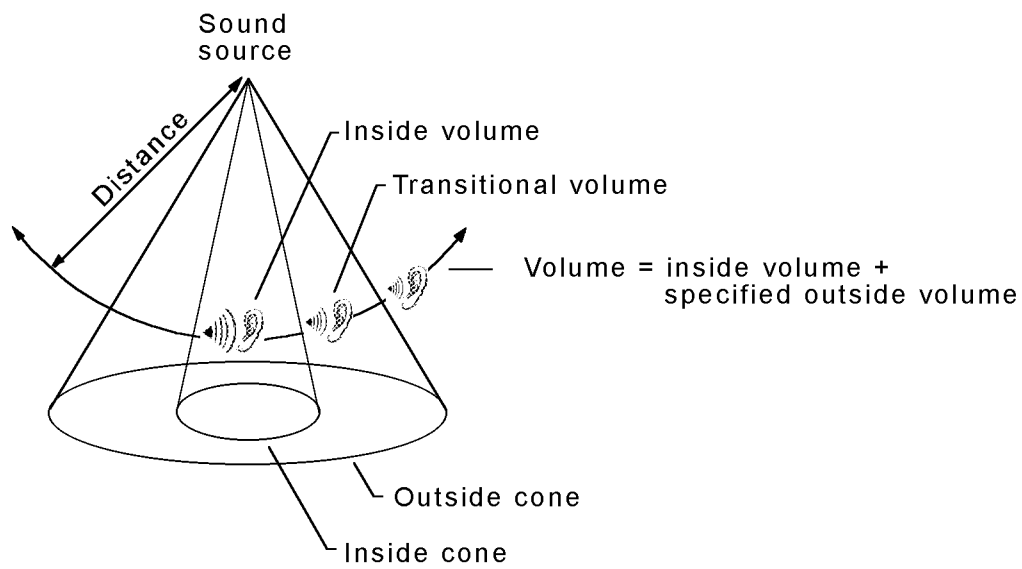
DirectSound の□□□□□□□□

のの(DirectSound

0 ののの

の)ののの-10,000 のの

□



IDirectSound3Dbuffer

□□□□□□□□ □□□□□□□□ ∞ ∞ 0

∞ ∞

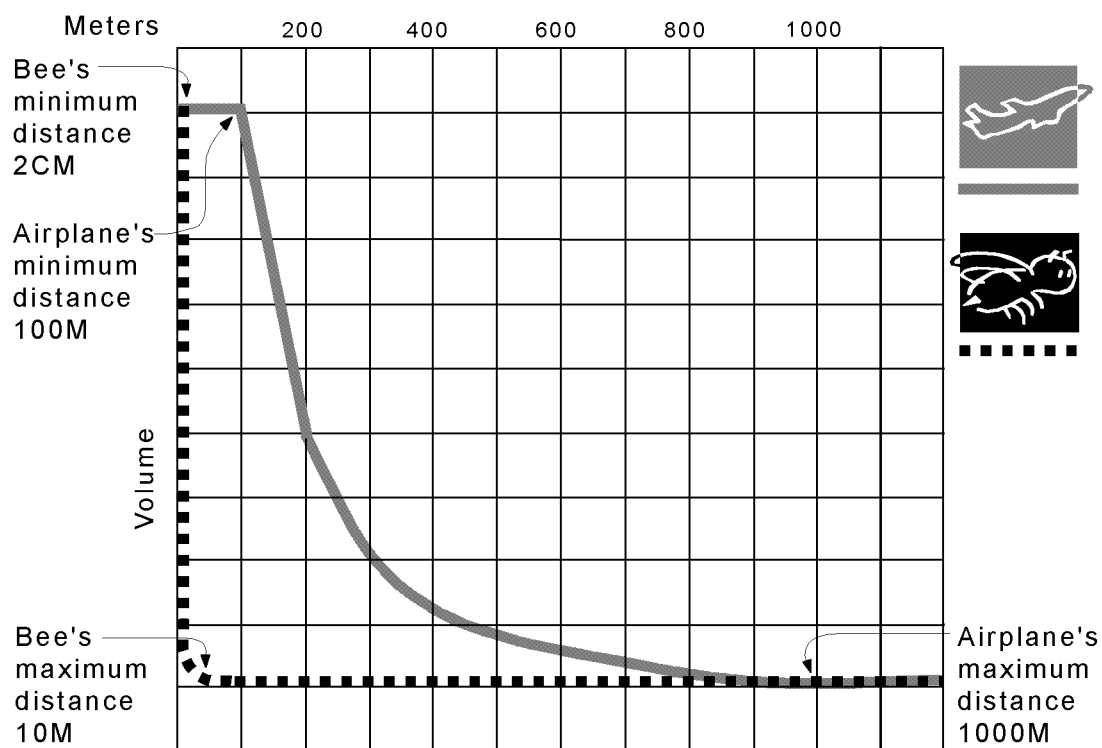
□'←, □'←

(0) ∞ ∞ ∞ ∞ ∞

(16 ∞ ∞ ∞ ∞) 100 ∞

2 ∞
200
∞ ∞

4



、 “

の 3D
の

3D の の 3D

3D の の 3D

の

DirectSound の

の の の の の
の の

3D

の

の

の の

Direct3D,, “ □

IDirectSound3DBuffer

IDirectSound3DListener

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 ののののの 0.30480006096 1

ffff%œ,ffff%œ

3D のの

DirectSound の 3D の

DirectSound**ffff**□**ffffŠ—**

のの DirectSound

- *IDirectSoundffff ffff*
- *IDirectSound3DBuffer fff ffff*
- *IDirectSound3DListener fff ffff*
- *IDirectSoundBuffer fff ffff*

IDirectSoundffff□**ffff**

DirectSound DirectSoundBuffer

DirectSound *IDirectSoundBuffer*

IDirectSound のの

のののののののの

- *ffff, ”*
- *ffff,*
- *ff f,Š< ’*
- *f fffffffŠ—*

ffff,□”

DirectSoundCreate

DirectSound

IDirectSound::GetCaps

ffff,□□

DirectSoundCreate

DirectSound

IDirectSound::CreateSoundBuffer

IDirectSound::DuplicateSoundBuffer

IDirectSound::SetCooperativeLevel

の

DSSCL_NORMAL の

ff **f**, **Š** **<** **□** **'**

IDirectSound

の

IDirectSound::GetSpeakerConfig

IDirectSound::SetSpeakerConfig

f **□** **ffffff** **Š** **—**

IDirectSound::Compact

の

IDirectSound3DBuffer **fff** **□** **ffff**

IDirectSound3DBuffer の

3D の

の **IDirectSound3DBuffer** のの

- *IDirectSound3DBuffer* **fff** **ffffff**, **Ž**“
- **ffffff** **f**“
- **'<—'**, **'<—'**
- **' f f**
- **^**, “
- **ffff** **ffffff** **f f**

IDirectSound3DBuffer **fff** **□** **fffffff**, **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)) {
    // , , ffff, 3Dfff□f, □' , ,
    .
    .
    .
}
```

DirectSound 3D の

3D の

3D

Note If you are using DirectSound, make sure that the DSBCAPS_CTRL3D and DSBCAPS_CTRLPAN flags are set in the DSB's capabilities.

ffffff ☐ ***f'*** ☐
3D の

\mathcal{O}
IDirectSound3DBuffer
 \mathcal{O}

IDirectSound3DBuffer::GetAllParameters

3D



⑦ IDirectSound3DBuffer::SetMinDistance
 IDirectSound3DBuffer::GetMinDistance
 ⑧ IDirectSound3DBuffer::SetMaxDistance
 IDirectSound3DBuffer::GetMaxDistance

$$\lceil \lceil f \rceil \rceil f$$

())))

3D \mathcal{O} **IDirectSound3DBuffer::SetMode**
 \mathcal{O} *dwMode* \mathcal{O}

^, “
;

3D	3D	⑦
DirectSound3DBuffer::SetPosition		
IDirectSound3DBuffer::GetPosition		

DirectSound
IDirectSound3DBuffer::SetVelocity
IDirectSound3DBuffer::GetVelocity


$f f f f \square f f f f f f f f \square f \square f$
 3D
IDirectSound3DBuffer
 ∞

IDirectSound3DBuffer

ののの

• f , \check{S} , • \check{S}

• “• \check{S} • f f fff f

f □ f , \check{S} “,• \check{S}

の

IDirectSound3DBuffer::SetConeAngles

IDirectSound3DBuffer::GetConeAngles

IDirectSound3DBuffer::SetConeOrientation

IDirectSound3DBuffer::GetConeOrientation

の 360

の

“•□ \check{S} • f □ f □ fff □ f

の

ののののの 100dB

の

0

の **IDirectSound3DBuffer::SetConeOutsideVolume**

IDirectSound3DBuffer::GetConeOutsideVolume

の

(の **IDirectSoundBuffer::GetVolume**

)

IDirectSound3DListener fff □ $ffff$

3D

3D の

の

IDirectSound3DListener の

3D

の

IDirectSound3DListener

のの

の

• *IDirectSound3DListener fff $ffff$ $ffff$, \check{Z}* “

• *$ffffff$ f* ‘

• ‘,’,’

• ‘——’

• *$ffff$ —*‘

• *fff ,^’,“*

• *fff ,• \mathcal{E}*

• *f fff —*‘

IDirectSound3DListener fff □ $ffff$ □ $ffff$, \check{Z} “

IDirectSound3DListener

3D

IDirectSound::CreateSoundBuffer

DSBUFFERDESC

の dwFlags DSBCAPS_CTRL3D
IDirectSoundBuffer::QueryInterface
IDirectSound3DListener の

```
// DSBCAPS_CTRL3D, , lpDsbPrimary, , ,  
  
hr = lpDsbPrimary->QueryInterface(IID_IDirectSound3DListener,  
    &lpDs3dListener);  
  
if (SUCCEEDED(hr)) {  
    // 3D , ,  
    .  
    .  
    .  
}
```

fff fff f'

の 3D の

IDirectSound3DListener

の

IDirectSound3DListener::GetAllParameters

IDirectSound3DListener::SetAllParameters

, , ,

3D

3D

の

の dwApply

DS3D_DEFERRED の

IDirectSound3DListener::CommitDeferredSettings

i

Note , , , , IDirectSound3DListener::CommitDeferredSettings, , ,
 , “ , , DS3D_IMMEDIATE fff, , , , , , , fff, ,
 , fff “, (1,2,3), , , , , fff, , fff “, (4,5,6), , , , , fff “, (4,5,6)
 , , , ,
DirectSound3DListener::CommitDeferredSettings fff, , , “, (4,5,6)
 , , ,

←

DirectSound の

の

IDirectSound3DListener::SetDistanceFactor

0.30480006096

(の 1

flDistanceFactor

) の

の IDirectSound3DListener::GetDistanceFactor

DS3D_DEFAULTDISTANCEFACTOR (1.0) の 1

(3.0, 7.2, -20.9) の 3m

7.2m 20.9m 6m 14.4m 41.8m 2.0

ffff□—‘

DirectSound 3D

DirectSound

10 の

IDirectSound3DListener::SetDopplerFactor

0 10 の0 の

の1

2

2

3D

IDirectSound3DListener::GetDopplerFactor

fff□,^’,“

IDirectSound3DListener::SetPosition

IDirectSound3DListener::GetPosition

3D の

の **IDirectSound3DListener::SetVelocity**

IDirectSound3DListener::GetVelocity

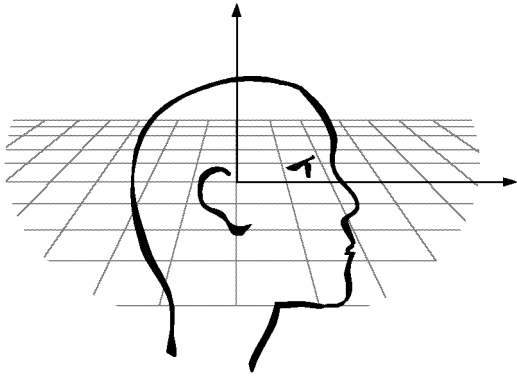
のの

fff□,•œ

の 3D のののDirectSound

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)

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

IDirectSoundBuffer

- Š—
- ffffŠ<Š—
- •,Ž“
- fffŠ—

IDirectSoundBuffer::SetCurrentPosition

ffffŠŠ—**IDirectSoundBuffer::GetVolume****IDirectSoundBuffer::SetVolume**

∞∞

IDirectSoundBuffer::GetFrequency**IDirectSoundBuffer::SetFrequency**

∞

∞

IDirectSoundBuffer::GetPan**IDirectSoundBuffer::SetPan****□•Ž“****IDirectSoundBuffer::GetCaps****DirectSoundBuffer****IDirectSoundBuffer::GetStatus****IDirectSoundBuffer::GetFormat**

∞

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

Note fffff ffff ffff, ,,, ,Ž,Ž',,, ,•Ž,ffffff ffff,•—
 ,, ,ff fff, <ffff, ,•—,,

fffŠ—**IDirectSoundBuffer::Restore****DirectSoundBuffer****IDirectSoundBuffer::Restore ∞∞**

∞∞

DirectSound,Ž—**DirectSound ∞**1 **DirectSoundCreateŠ** ,Ž, ,,DirectSoundffffff, ,,

2

IDirectSound::SetCooperativeLevelffff,Ž, ,,<fff,Ž',,, ,fff fff
 ,, %o^fff,,DSSCL_NORMAL,Ž—,,

3 **IDirectSound::CreateSoundBuffer**ffff,Ž—

,,ffffff ffff, ,, **DSBUFFERDESC** “,’,, ,fff,ffffff ffff,,,,,
 Ž',•—,, fffff,ffffff ffff, ,,,,,,,

4 fffff ffff,f f,“ , f f—

^,ffff,Ž“,,, **IDirectSoundBuffer::Lock**ffff,Ž—, ,ffff,f f,fff,,, **IDirectSoundBuffer::Unlock**ffff,Ž—,,5 **IDirectSoundBuffer::Play**ffff,Ž—,,ffffff ffff, ,,

```

6 ffff fff,fff, , —
  ,, DirectSoundBufferfff, IDirectSoundBuffer::Stopfff,Ž—
  ,,,,fff,Ž,,
7 fffff ffff,%o*,
8 DirectSoundffffff,%o*,

```

○

-

```

ffff ffff, , IDirectSoundBuffer::SetFormatfff,Œ, , ffff ff
ff ffff, —
ŒŽ, ', , ',', ffff fff,<fff,DSSCL_PRIORITY, ',,,,•—,,

```

- fffff ffff ffff, , IDirectSoundBuffer::Playfff,Ž—

,,,ffff, , , , ,fffff ffff, ,,,,,, fffff ffff,,, ,,,,,,•

 , ,“ ,CPUff ,,,, ”,, fffff ffff, ,,,,,ŠŽŽŠ,' ,

○DirectSound

- DirectSoundffffff,
- CoCreateInstance,Ž—,,DirectSoundffffff,
- f ffff ”, %o
- ffff ffff,
- ffff ffff,, ,,,
- DirectSoundfff ,Ž—
- ffff fff ,Ž—
- ^ wave,Ž—

DirectSoundffffff,□□

DirectSound

DirectSoundCreate

NULL GUID

IDirectSound::SetCooperativeLevel

```

LPDIRECTSOUND lpDirectSound;
if(DS_OK == DirectSoundCreate(NULL, &lpDirectSound,
    NULL)) {
    // □□,Œ,□□
    lpDirectSound->lpVtbl->SetCooperativeLevel(lpDirectSound,
        hwnd, DSSCL_NORMAL);
    // .
    // . DirectSound,ffff,,,,,f□f,,,,"',,
    // .
} else {
    // □□,Ž",,□□
    // .
    // .
    // .
}

```

DirectSoundEnumerate**DSEnumCallback**

```
typedef struct {
    // GUIDs, <%, ,
    // ffff, ffffffff, Ž-, <%, ,
} APPINSTANCEDATA, *LPAPPINSTANCEDATA;
BOOL AppEnumCallbackFunction(
    LPGUID lpGuid,
    LPTSTR lpstrDescription,
    LPTSTR lpstrModule,
    LPVOID lpContext)
{
    LPAPPINSTANCEDATA lpInstance = (LPAPPINSTANCEDATA)
    lpContext;
    // lpInstance[]'', GUID, ff[], ,
    // Strcpy<[]Ž-, lpInstance[]'',
    return TRUE; // -<, ', ,
}
```

の

DirectSound

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

    // DirectSoundCreate, ff[], , , ,
    // •', ffff, , •-, , ,

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

*lpGuid***DirectSoundCreate**

の

CoCreateInstance, Ž—
,, DirectSound ffffffff, [] []

CoCreateInstance

DirectSound

```
1 Š ,NULL,Ž',, COM, Š%, ,
    if (FAILED(CoInitialize(NULL)))
```

```
return FALSE;
```

```
2 DirectSoundCreateŠ ,Ž—,,, CoCreateInstance,,,
  IDirectSound::Initializeffff,Ž—,,DirectSoundffffff, ,,
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();
```

f ffff", %o

DirectSound DirectSound の

IDirectSound::GetCaps
DSCAPS □□□

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

DSCAPS

の

の

の

の IDirectSound::GetCaps

Windows

ffff **ffff**,

の の の の の

- Š–“,ffff ffff,
- Œffff
- fffffff ffff ffff,fff f ffff ffff
- f ffff ffff ffff,ffffff ffff ffff
- fffff ffff ffff,ffffff ffff ffff

Š–“,ffff **ffff**, **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;
    // waveff[]ff[]'',[],,
    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);
    // fffff,ffff[]f,-<,,(ff[]ff[]f[]ž")[]
    dsbdesc.dwFlags = DSBCAPS_CTRLDEFAULT;
    // 3•,ffff
    dsbdesc.dwBufferBytes = 3 * pcmwf.wf.nAvgBytesPerSec;
    dsbdesc.lpwfxFormat = (LPWAVEFORMATEX) &pcmwf;
    // ffff,[],,
    hr = lpDirectSound->lpVtbl->CreateSoundBuffer(lpDirectSound,
        &dsbdesc, lplpDsb, NULL);

```

□ **Œffff**

DirectSound

DSBCAPS_CTRLVOLUME

fffff ☐ *ffff* ☐ *ffff*, *fff* ☐ *f* ☐ *ffff* ☐ *ffff*

PCI の の の の 1 の

$f \square f f f f \square f f f f \square f f f f, f f f f f f \square f f f f \square f f f f$

 \mathcal{O}

DSBCAPS_LOCHARDWARE

```

    IDirectSoundBuffer::GetCaps
    DSB_CAPS dwFlags DSB_CAPS_LOCHARDWARE
    DSB_CAPS_LOCSOFTWARE

```

```

    ffffffff, ffffffff, ffffffff, ffffffff, ffffffff, ffffffff

```

DSBUFFERDESC

DirectSound の の の の

```

    IDirectSoundBuffer::SetFormat
    IDirectSoundBuffer::Lock
    IDirectSoundBuffer::GetCurrentPosition

```

DirectSound の

DSBPLAY_LOOPING

```

    IDirectSound

```

```

BOOL AppCreateWritePrimaryBuffer(
    LPDIRECTSOUND lpDirectSound, LPDIRECTSOUNDBUFFER *lplpDsb,
    LPDWORD lpdwBufferSize, HWND hwnd)
{
    DSBUFFERDESC dsbdesc;
    DSB_CAPS dsbcaps;
    HRESULT hr;
    // wave format, 16-bit, 2 channels
    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, 16-bit, 2 channels
    memset(&lplpDsb, 0, sizeof(DSBUFFERDESC)); // 0, 16-bit
    dsbdesc.dwSize = sizeof(DSBUFFERDESC);
    dsbdesc.dwFlags = DSB_CAPS_PRIMARYBUFFER;
    // ffffffff, ffffffff, ffffffff, ffffffff, ffffffff, ffffffff
    dsbdesc.dwBufferBytes = 0;
    dsbdesc.lpwfxFormat = NULL; // ffffffff, ffffffff, ffffffff, ffffffff, ffffffff, ffffffff

    // 16-bit, 2 channels, 22050 Hz
    hr = lpDirectSound->lpVtbl->SetCooperativeLevel(lpDirectSound,
        hwnd, DSSCL_WRITEPRIMARY);
    if (DS_OK == hr) {

```



```
// BUFFERLOST,*,,,,ffff,fff,
if(DSEERR_BUFFERLOST == hr) {
    lpDsb->lpVtbl->Restore(lpDsb);
    hr = lpDsb->lpVtbl->Lock(lpDsb, dwOffset, dwSoundBytes,
        &lpvPtr1, &dwAudio1, &lpvPtr2, &dwAudio2, 0);
}
if(DS_OK == hr) {
    // ffff,.,,,,
    CopyMemory(lpvPtr1, lpbSoundData, dwBytes1);
    if(NULL != lpvPtr2) {
        CopyMemory(lpvPtr2, lpbSoundData+dwBytes1, dwBytes2);
    }
    // DirectSound,f,f,*,
    hr = lpDsb->lpVtbl->Unlock(lpDsb, lpvPtr1, dwBytes1, lpvPtr2,
        dwBytes2);
    if(DS_OK == hr) {
        //
        return TRUE;
    }
}
// ffffff%,.,,,-,,"
return FALSE;
}
```

DirectSoundfff,Ž—

DirectSound の の IDirectSoundBuffer

の

IDirectSoundBuffer::Lock **IDirectSoundBuffer::Unlock**
 IDirectSoundBuffer::Play
 IDirectSoundBuffer::Stop

IDirectSoundBuffer::Play

の
の 0

IDirectSoundBuffer::SetCurrentPosition

IDirectSoundBuffer::Play の
の ()
の **IDirectSoundBuffer::Play**

 Win32 **SetTimer**
SetEvent DirectSound DirectDraw

DirectSound

DSBCAPS_STATIC

DirectSound の

DirectSound の
の

DirectSound の wave の

の IDirectSoundBuffer::SetFormat
の の
DSSCL_PRIORITY のの DirectSound IDirectSoundBuffer::SetFormat

ffff ffff,Ž—
の DirectSound の

DirectSound

のののののの

DSSCL_WRITEPRIMARY
IDirectSoundBuffer::Lock

IDirectSoundBuffer::Unlock
IDirectSoundBuffer::Play
IDirectSoundBuffer::Play のの
DirectSound
DSBPLAY_LOOPING

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;  
    // ffff,Ž  
    hr = lpDsbPrimary->lpVtbl->Lock(lpDsbPrimary, dwOldPos,  
dwDataBytes,  
    &lpvPtr1, &dwBytes1, &lpvPtr2, &dwBytes2, 0);  
    // DSERR_BUFFERLOST,ffff,fff,  
    if(DSERR_BUFFERLOST == hr) {  
        lpDsbPrimary->lpVtbl->Restore(lpDsbPrimary);  
        hr = lpDsbPrimary->lpVtbl->Lock(lpDsbPrimary, dwOldPos,  
dwDataBytes, &lpvPtr1, &dwBytes1, &lpvPtr2, &dwBytes2,  
0);  
    }  
}
```

```

    }
    if(DS_OK == hr) {
        // ・・・ffff,f,f,ffff,,
        CustomMixer(lpAppStreamInfo, lpvPtr1, dwBytes1);
        *lpdwNewPos = dwOldPos + dwBytes1;
        if(NULL != lpvPtr2) {
            CustomMixer(lpAppStreamInfo, lpvPtr2, dwBytes2);
            *lpdwNewPos = dwBytes2; // Š,-,,,,
        }
        // DirectSound,f,f,・,
        hr = lpDsbPrimary->lpVtbl->Unlock(lpDsbPrimary, lpvPtr1,
            dwBytes1, lpvPtr2, dwBytes2);
        if(DS_OK == hr) {
            // ㊦
            return TRUE;
        }
    }
    // fff,,fff%□,ž”
    return FALSE;
}

```

^□wave,ž—

の DirectSound

wave

PCM(pulse-coded modulation)
(ACM)

Win32 SDK

の

ACM

DirectSound**ffffff**

Š□

DirectSoundCreate

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

IDirectSound の

- DS_OK の
- DSERR_ALLOCATED
- DSERR_INVALIDPARAM
- DSERR_NOAGGREGATION
- DSERR_NODRIVER
- DSERR_OUTOFMEMORY

lpGuid

ffffffff,Ž•,GUID,,ffff□
,,fff f,’,DirectSoundEnumerate,,,•,,GUIDs,,, ,,,,ffffff,ffff,—<,,,,,
NULL,,,

ppDS

„Š „%“,, ,,,DirectSoundffffff,,ffff

pUnkOuter

—’<^ ,,’,NULL,,,,,,,

DirectSound IDirectSound::SetCooperativeLevel

IDirectSound::GetCaps, IDirectSound::SetCooperativeLevel

DirectSoundEnumerate

BOOL DirectSoundEnumerate(
LPDSENUMCALLBACK lpDSEnumCallback, LPVOID lpContext);

DirectSound

- Œ,,,,_, Ž”,,,DSERR_INVALIDPARAM,ff f f,•,

lpDSEnumCallback

DSEnumCallbackŠ „ffff „ffff,ffff f,,,,,,,DirectSoundffffff,,
Œ, ,,,

lpContext

f f’<ffffff,,ffff „ffffff, —< ffffŠ ,Œ, ,,,,, „Š „“,,

DSEnumCallback

f ffffŠ

DSEnumCallback

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

DirectSound

• ffff,—,‘, ,TRUE —,, ,FALSE,•,

lpGuid

—‘,ffff,Ž•,,GUID,,ffff ,’, ffff,DirectSoundfffff, ,,,,DirectS
oundCreateŠ ,“,,

lpstrDescription

DirectSoundffff,ffff< ,—,, NULL, ,,•Ž—,,ffff

lpstrModule

,,ffff,“,,DirectSoundffff,fff f—,Ž’,, NULL, ,,•Ž—,,ffff

lpContext

ffff fff’<f f,,ffff ,,,,f ffffŠ ,“,,

lpstrDescription *lpstrModule* ののののの

DirectSoundEnumerate

IDirectSound

DirectSound IDirectSound のののののの IDirectSound

IDirectSound のの

fffŠ,“,

Compact

Initialize

ffff

CreateSoundBuffer

DuplicateSoundBuffer

SetCooperativeLevel

ffff ”

GetCaps

ff f ’

GetSpeakerConfig

SetSpeakerConfig

Unknown

IDirectSound::Compact

- `DS_OK`, `DS_ERR_INVALIDPARAM`, `DS_ERR_PRIOLEVELNEEDED`, `DS_ERR_UNINITIALIZED`

IDirectSound, **IDirectSound::SetCooperativeLevel**

IDirectSound::CreateSoundBuffer

DirectSoundBuffer

- `DS_OK`, `DS_ERR_ALLOCATED`, `DS_ERR_BADFORMAT`, `DS_ERR_INVALIDPARAM`, `DS_ERR_NOAGGREGATION`, `DS_ERR_OUTOFMEMORY`, `DS_ERR_UNINITIALIZED`, `DS_ERR_UNSUPPORTED`

```
lpDSBufferDesc
    ,,,ffff ffff, < ,Š,,,,,DSBUFFERDESC “,ffff
lpDirectSoundBuffer
    <,DirectSoundBufferffffff,,ffff ffff, ,,,,, ,NULL,,
pUnkOuter
    -‘<^ ,,,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 lpDsbDuplicate);
```

DirectSoundBuffer

- DS_OK, DSERR_ALLOCATED, DSERR_INVALIDCALL, DSERR_INVALIDPARAM, DSERR_OUTOFMEMORY, DSERR_UNINITIALIZED

DSERR_ALLOCATED**DSERR_INVALIDCALL****DSERR_INVALIDPARAM****DSERR_OUTOFMEMORY****DSERR_UNINITIALIZED***lpDsbOriginal*

- DirectSoundBufferfffff,ffff

lpDsbDuplicate

DirectSoundfffff,ffff

の

のの

の

IDirectSound, **IDirectSound::CreateSoundBuffer**

IDirectSound::GetCaps

HRESULT GetCaps(LPDSCAPS lpDSCaps);

DirectSound

- DS_OK, DSERR_GENERIC, DSERR_INVALIDPARAM, DSERR_UNINITIALIZED

lpDSCaps

DSCAPS

DirectSoundCreate, DSCAPS, IDirectSound

IDirectSound::GetSpeakerConfig

HRESULT GetSpeakerConfig(LPDWORD lpdwSpeakerConfig);

DirectSound

- DS_OK, DSERR_INVALIDPARAM, DSERR_UNINITIALIZED

lpdwSpeakerConfig

DSSPEAKER_HEADPHONE, DSSPEAKER_MONO, DSSPEAKER_QUAD, DSSPEAKER_STEREO, DSSPEAKER_SURROUND

IDirectSound, IDirectSound::SetSpeakerConfig

IDirectSound::Initialize

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

CoCreateInstance DirectSound

- DS_OK, DSERR_ALREADYINITIALIZED, DSERR_GENERIC, DSERR_INVALIDPARAM, DSERR_NODRIVER

lpGuid

```
ff ffff ffffff(GUID)
,,ffff ,,Z,Z, ,,DirectSoundfffff,CE ,,,,ffffffffff,Z',, NULL, ,f
ffff ffff ffff,“,,
```

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);
```

- DS_OK, DSERR_ALLOCATED, DSERR_INVALIDPARAM, DSERR_UNINITIALIZED, DSERR_UNSUPPORTED

hwnd

```
ffff fff,fffff ffff
```

dwLevel

```
—<,,<'fff Z',Z',,,
```

DSSCL_EXCLUSIVE

```
ffff fff,“fff, ',, “—ff ff,“,,, ,ffff fff,,—^—
,,,,,,(DSBCAPS_GLOBALFOCUSfff ',ffff fff,,ffff,ff f,,)
,,fff,,, ffff fff, DSSCL_PRIORITYfff,,,,,“CE,—
, ,IDirectSoundBuffer::SetFormatffff,CE, ,Z',,,, DirectSoun
```

d,f ffffŒŽ,ffff,,(,, DirectSound, ',,—
fff,Š,,, ,,wave,ffff,,)

DSSCL_NORMAL

ffff fff, •,‹ ' ',,, ,ff f,ffffffffff,ff f<—
, ,,, ,,,,ffff fff,,,fff, ',,,

DSSCL_PRIORITY

ffff fff,—
fff, ',,, ,fff,ffff fff, IDirectSoundBuffer::SetFormat,,IDirectSound::Compactffff,Œ, ,,,,,,

DSSCL_WRITEPRIMARY

,,, ^,fff,,, ffff fff,ffffff ffff ffff,, ,,,ffff,—
,,,, fffff ffff ffff, ^ ,,,

⑦

⑦

DSSCL_NORMAL ⑦

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,ff f f,•,

DSERR_INVALIDPARAM

DSERR_UNINITIALIZED

dwSpeakerConfig

DirectSoundffffff,ff f ' ^%o,'Ž',,,

DSSPEAKER_HEADPHONE

ff f,ffffff,,,

DSSPEAKER_MONO

ff f,ffff,,,

DSSPEAKER_QUAD

ff f,4ffffff,,,

DSSPEAKER_STEREO

ff f,ffff,,(fffff')

DSSPEAKER_SURROUND

ff f,ffffff ffff,,,

IDirectSound, **IDirectSound::GetSpeakerConfig**

IDirectSound3DBuffer

の 3D の	
IDirectSound3DBuffer <i>IDirectSound3DBuffer</i>	の の の の の の
IDirectSound3DBuffer の の	
ffffff f	GetAllParameters
‘	SetAllParameters
←	GetMaxDistance GetMinDistance SetMaxDistance SetMinDistance
‘ f f	GetMode SetMode
~	GetPosition SetPosition
ffff ffffffff f f	GetConeAngles GetConeOrientation GetConeOutsideVolume SetConeAngles SetConeOrientation SetConeOutsideVolume
“	GetVelocity SetVelocity
の COM IUknown	IDirectSound3DBuffer IUnknown の の
AddRef	
QueryInterface	
Release	

IDirectSound3DBuffer::GetAllParameters

HRESULT GetAllParameters(LPDS3DBUFFER lpDs3dBuffer);

の の 3D

- $\mathbb{E},,,\text{DS_OK}$ \check{Z} ”,,,DSERR_INVALIDPARAMff f f,•,

lpDs3dBuffer

DS3DBUFFER “,,ffff ,, “,ffff ffff,3D“ ,,,, •,Ž,,,

IDirectSound3DBuffer::GetConeAngles

HRESULT GetConeAngles(
LPDWORD lpdwInsideConeAngle, LPDWORD lpdwOutsideConeAngle);

∞

- $\mathbb{E},,,\text{DS_OK}$ \check{Z} ”,,,DSERR_INVALIDPARAMff f f,•,

lpdwInsideConeAngle *lpdwOutsideConeAngle*

ffff ffffffff f f,“•Š,Š•Š,Ž,• ,,ffff

IDirectSound3DBuffer::GetConeOrientation

HRESULT GetConeOrientation(LPD3DVECTOR lpvOrientation);

∞

- $\mathbb{E},,,\text{DS_OK}$ \check{Z} ”,,,DSERR_INVALIDPARAMff f f,•,

lpvOrientation

D3DVECTOR “,,ffff ,, “,ffff ffffffff f f, \mathbb{E} ,• \mathbb{E} ,Ž,,, fff
f •,ffff f f,’ ,•,

IDirectSound3DBuffer::SetConeAngles,
IDirectSound3DBuffer::SetConeOutsideVolume

IDirectSound3DBuffer::GetConeOutsideVolume

HRESULT GetConeOutsideVolume(LPLONG lplConeOutsideVolume);

∞

- $\mathbb{E},,,\text{DS_OK}$ \check{Z} ”,,,DSERR_INVALIDPARAMff f f,•,

lplConeOutsideVolume

ffff, \mathbb{E} ,f fŠ•fff f,Ž,• ,,ffff

0() -10,000()∞

DirectSound

∞

IDirectSoundBuffer::SetVolume

IDirectSound3DBuffer::GetMaxDistance

HRESULT GetMaxDistance(LPD3DVALUE lpflMaxDistance);

∞

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

lpflMaxDistance

∞ , '← ',Ž,• ,ffff

∞

□□□□□□□□□□□□□□□□

IDirectSound3DBuffer::GetMinDistance,
IDirectSound3DBuffer::SetMaxDistance

IDirectSound3DBuffer::GetMinDistance

HRESULT GetMinDistance(LPD3DVALUE lpflMinDistance);

∞

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

lpflMinDistance

∞ , '← ',Ž,• ,ffff

1.0 ∞

1.0 ∞

□□□□□□□□□□□□□□□□

IDirectSound3DBuffer::SetMinDistance,
IDirectSound3DBuffer::GetMaxDistance

IDirectSound3DBuffer::GetMode

HRESULT GetMode(LPDWORD lpdwMode);

3D ∞

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

lpdwMode

∞ , f f,Ž,• ,ffff Ž',•,

DS3DMODE_DISABLE

3Dffff —,—∞,,, ffff,fff ,“,’ ,•,,,,,Š,,

DS3DMODE_HEADRELATIVE

ffffff f(“ “ •∞)

,fff ,fff f,“Š∞,,, ,f f,, ffff, ‘fff f,fff fff f,• ,,,,,
Ž““ , ,,,, “fff f,•%0,,,,,,

DS3DMODE_NORMAL

• , — ,,,ffffff f,,,,,

IDirectSound3DBuffer::GetPosition

HRESULT GetPosition(LPD3DVECTOR lpvPosition);

∞∞

- Ą,,,DS_OK Ź”,,,DSERR_INVALIDPARAMff f f,•,

lpvPosition

D3DVECTOR “,,ffff ,, “,ffff ffff’,ffff fffff,Ź,,,

IDirectSound3DBuffer::GetVelocity

HRESULT GetVelocity(LPD3DVECTOR lpvVelocity);

∞∞

- Ą,,,DS_OK Ź”,,,DSERR_INVALIDPARAMff f f,•,

lpvVelocity

D3DVECTOR “,,ffff ,, “,ffff ffff,Ą “,Ź,,,

∅

IDirectSound3DBuffer::SetPosition,
IDirectSound3DBuffer::SetVelocity

IDirectSound3DBuffer::SetAllParameters

HRESULT SetAllParameters(
LPDS3DBUFFER lpDs3dBuffer, DWORD dwApply);

∞∞∞

DS3DBUFFER

∅ 3D

- Ą,,,DS_OK Ź”,,,DSERR_INVALIDPARAMff f f,•,

lpDs3dBuffer

DS3DBUFFER “,,ffff ,, “,ffff ffff,3D“ ,Ź,,,

dwApply

,’fffff,Ź,’ Ź,’Ź,,,

DS3D_DEFERRED

DS3D_IMMEDIATE

ffff fff,IDirectSound3DListener::Com
mitDeferredSettingsffff,Ą, ,, , ,,,

• , ,’• ,,, “, ĄŹ,Ź ,,,,%0”

,’Ź, ,, ffff,,,,,3Dffff ffff,’,,3D •
, ĄŹ, ,

IDirectSound3DBuffer::SetConeAngles

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

○○○

- $\mathbb{E},,,\text{DS_OK}$ $\checkmark,,,,\text{DSERR_INVALIDPARAM}$ ff f f, \bullet ,

$dwInsideConeAngle$ $dwOutsideConeAngle$
 $ffff$ $ffffffff$ f $f, \bullet \checkmark, \checkmark, \checkmark$

$dwApply$
 $, ,,,ffff, \checkmark, \checkmark \checkmark, \checkmark, \checkmark, ,,,$

DS3D_DEFERRED

$ffff$ $fff, \text{IDirectSound3DListener::CommitDeferredSettings}$ $ffff, \mathbb{E}, ,,, , ,,,$
 $\bullet , , \bullet ,,, \text{“} \mathbb{E} \checkmark, \checkmark ,,,, \% \text{”}$

DS3D_IMMEDIATE

$, \checkmark, , , ffff, ,,, , 3D ffff ffff, \text{“}, 3D \bullet$
 $, \mathbb{E} \checkmark, ,$

○ 0 () 360 () 360

IDirectSound3DBuffer::GetConeOutsideVolume,
IDirectSound3DBuffer::SetConeOutsideVolume

IDirectSound3DBuffer::SetConeOrientation

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

○○○

- $\mathbb{E},,,\text{DS_OK}$ $\checkmark,,,,\text{DSERR_INVALIDPARAM}$ ff f f, \bullet ,

x, y, z
 $\text{D3DVALUE}, fff, \checkmark, ,,, \text{“}, ffff f f, \bullet \mathbb{E} ffff, \bullet, \bullet$,

$dwApply$
 $, ,,,ffff, \checkmark, \checkmark \% \text{“}, \checkmark, \checkmark, ,,,$

DS3D_DEFERRED

$ffff$ $fff, \text{IDirectSound3DListener::CommitDeferredSettings}$ $ffff, \mathbb{E}, ,,, , ,,, \bullet$
 $, , \bullet ,,, \text{“} \mathbb{E} \checkmark, \checkmark ,,,, \% \text{”}$

DS3D_IMMEDIATE

$, \checkmark, , , ffff, ,,, , 3D ffff ffff, \text{“}, 3D \bullet$
 $, \mathbb{E} \checkmark, ,$

IDirectSound3DBuffer::GetConeOrientation $lpvOrientation$ ○○
 (0,0,1)

IDirectSound3DBuffer::SetConeAngles,
IDirectSound3DBuffer::SetConeOutsideVolume

IDirectSound3DBuffer::SetConeOutsideVolume

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

○○

- E,,,DS_OK Ž”,,,DSERR_INVALIDPARAMff f f,•,

lConeOutsideVolume

ffff ffff,f fŠ•fff f,100ffff dB ,”^Ž’,, ”’%”””^0(E ,,) ,,-10,000(-‰),,,,

dwApply

, ,,,,fffff,Ž,’ ‰‰,’Ž,,,

DS3D_DEFERRED

ffff fff,IDirectSound3DListener::Com
mitDeferredSettingsffff,E, ,,, ’, ,,,,
• , ,’• ,,, ”“, EŽ,Ž ,,,,‰”

DS3D_IMMEDIATE

,‘Ž, , ffff,,,,,3Dffff ffff,‘,,3D •
, EŽ, ,

DirectSound

○○

IDirectSoundBuffer::SetVolume

IDirectSound3DBuffer::SetMaxDistance

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

○

- E,,,DS_OK Ž”,,,DSERR_INVALIDPARAMff f f,•,

flMaxDistance

,, ’←’

dwApply

, ,,,,fffff,Ž,’ ‰‰,’Ž,,,

DS3D_DEFERRED

ffff fff,IDirectSound3DListener::Com
mitDeferredSettingsffff,E, ,,, ’, ,,,,
• , ,’• ,,, ”“, EŽ,Ž ,,,,‰”

DS3D_IMMEDIATE

,‘Ž, , ffff,,,,,3Dffff ffff,‘,,3D •
, EŽ, ,

○

□□□□□□□□□□□□□□□□□□

IDirectSound3DBuffer::GetMaxDistance,
IDirectSound3DBuffer::SetMinDistance

IDirectSound3DBuffer::SetMinDistance

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

⑦

DS_OK

DSERR_INVALIDPARAM

flMinDistance

,, '←'

dwApply

', ,,,fffff,Ž,' ~%,',Ž,,

DS3D_DEFERRED

ffff fff, IDirectSound3DListener::Com
 mitDeferredSettingsffff,€, ,,, ', ,,,
 • , ',• ,,, "€, Ž,Ž ,,,,~%"

DS3D_IMMEDIATE

',Ž, , ffff,,,,,3Dffff ffff,€,3D •
 , €,Ž, ,

1.0(∞)

1.0

)

⑦

□□□□□□□□□□

IDirectSound3DBuffer::SetMaxDistance

IDirectSound3DBuffer::SetMode

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

3D

• €,,,DS_OK Ž",,,DSERR_INVALIDPARAMff f f,•,

dwMode

3Dffff —f f, ',, Ž',Ž,,

DS3DMODE_DISABLE

3Dffff —,€ ffff,fff ,“, ' ,,,•,,,,,Š,,

DS3DMODE_HEADRELATIVE

ffff fff f(^) “ •(€)
 ,fff ,fff f,“Š€, ,,, ,f f,, ffff, 'fff f,fff fff f,• ,,,
 Ž““, ,,,, “fff f,•,,,,,,

DS3DMODE_NORMAL

• , — ,,,fffff f f,,,,,

dwApply

', ,,,fffff,Ž,' ~%,',Ž,,

DS3D_DEFERRED

ffff fff, IDirectSound3DListener::Com
mitDeferredSettingsffff, Ě, ,,, ', ,,,,
• , ', • ,,, " ĚŽ, Ž ,,,, %o"

DS3D_IMMEDIATE

', Ž, ,,, ffff, ,,, 3Dffff ffff, ', 3D •
, ĚŽ, ,

IDirectSound3DBuffer::SetPosition

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

∞∞

IDirectSound3DListener::SetDistanceFactor

• Ě, ,,, DS_OK Ž" ,,, DSERR_INVALIDPARAMff f f, •,

x, y, z
D3DVALUE, fff, Ž, ' ,,, ,,, ^ffff, •, •,

dwApply
' , ,,, fffff, Ž, ' ^%o, ' Ž, ,,,

DS3D_DEFERRED

ffff fff, IDirectSound3DListener::Com
mitDeferredSettingsffff, Ě, ,,, ', ,,,,
• , ', • ,,, " ĚŽ, Ž ,,,, %o"

DS3D_IMMEDIATE

', Ž, ,,, ffff, ,,, 3Dffff ffff, ', 3D •
, ĚŽ, ,

IDirectSound3DBuffer::SetVelocity

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

∞∞

• Ě, ,,, DS_OK Ž" ,,, DSERR_INVALIDPARAMff f f, •,

x, y, z
D3DVALUE, fff, Ž, ' ,,, ,,, "ffff, •, •,

dwApply
' , ,,, fffff, Ž, ' ^%o, ' Ž, ,,,

DS3D_DEFERRED

ffff fff, IDirectSound3DListener::Com
mitDeferredSettingsffff, Ě, ,,, ', ,,,,
• , ', • ,,, " ĚŽ, Ž ,,,, %o"

DS3D_IMMEDIATE

', Ž, ,,, ffff, ,,, 3Dffff ffff, ', 3D •
, ĚŽ, ,

∅

**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
~	GetPosition SetPosition
f fff—‘	GetRolloffFactor SetRolloffFactor
““	GetVelocity SetVelocity

の COM IDirectSound3DBuffer
IUknown IUknown のの
AddRef
QueryInterface
Release

IDirectSound3Dlistener ::CommitDeferredSettings

HRESULT CommitDeferredSettings();

ののの

- $\mathbb{C}_{,,,DS_OK}$ $\check{Z}_{,,,DSERR_INVALIDPARAM}$ ff f f ,•,

のの

□□□□□□□□□□□□□□

IDirectSound3DListener::GetAllParameters

HRESULT GetAllParameters(LPDS3DLISTENER lpListener);

3D 3D のの

- $\mathbb{C}_{,,,DS_OK}$ $\check{Z}_{,,,DSERR_INVALIDPARAM}$ ff f f ,•,

lpListener

DS3DLISTENER “,,ffff ,, “,3Df ff,,3Dfff , \mathbb{C} , ‘ \check{Z} ,

DirectSound3DListener::SetAllParameters

IDirectSound3DListener::GetDistanceFactor

HRESULT GetDistanceFactor(LPD3DVALUE lpflDistanceFactor);

の

- $\mathbb{C}_{,,,DS_OK}$ $\check{Z}_{,,,DSERR_INVALIDPARAM}$ ff f f ,•,

lpflDistanceFactor

D3DVALUE fff ,• ,,ffff ,, \mathbb{C} ,←——‘,’ \check{Z} ,,,

IDirectSound3DListener::SetDistanceFactor

IDirectSound3DListener::GetDopplerFactor

HRESULT GetDopplerFactor(LPD3DVALUE lpflDopplerFactor);

の

- $\mathbb{C}_{,,,DS_OK}$ $\check{Z}_{,,,DSERR_INVALIDPARAM}$ ff f f ,•,

lpflDopplerFactor

D3DVALUE fff ,• ,,ffff ,, \mathbb{C} $ffff$ —‘,’ \check{Z} ,,,

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

- $\mathbb{E},,,\text{DS_OK}$ $\checkmark,,, \text{DSERR_INVALIDPARAM}$ ff f $f, \bullet,$

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**

- $\mathbb{E},,,\text{DS_OK}$ $\checkmark,,, \text{DSERR_INVALIDPARAM}$ ff f $f, \bullet,$

lpvPosition

D3DVECTOR “,,ffff fff , \mathbb{E} ,~,Ž,

IDirectSound3DListener::SetPosition

IDirectSound3DListener::GetRolloffFactor

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

の

- $\mathbb{E},,,\text{DS_OK}$ $\checkmark,,, \text{DSERR_INVALIDPARAM}$ ff f $f, \bullet,$

lpflRolloffFactor

D3DVALUE fff, \bullet ,ffff \mathbb{E} , f fff—‘,’Ž,,,

DS3D_DEFAULTROLLOFFFACTOR (1.0)

IDirectSound3DListener::SetRolloffFactor

IDirectSound3DListener::GetVelocity

HRESULT GetVelocity(LPD3DVECTOR lpvVelocity);

のの

- 成功した場合、DS_OK が返され、lpvVelocity は、リスナーの現在の速度を返す。

lpvVelocity

D3DVECTOR 型で、リスナーの現在の速度を返す。

のののの IDirectSound3DListener::SetPosition

(0,0,0)

IDirectSound3DListener::SetVelocity

IDirectSound3DListener::SetAllParameters

HRESULT SetAllParameters(
LPD3DLISTENER lpListener, DWORD dwApply);

DS3DBUFFER

3D ののの

- 成功した場合、DS_OK が返され、lpListener は、リスナーの現在の速度を返す。

lpListener

DS3DBUFFER 型で、リスナーの現在の速度を返す。

dwApply

、成功した場合、DS_OK が返され、lpListener は、リスナーの現在の速度を返す。

DS3D_DEFERRED

ffff fff, IDirectSound3DListener::CommitDeferredSettings ffff, 成功した場合、DS_OK が返され、lpListener は、リスナーの現在の速度を返す。

DS3D_IMMEDIATE

、成功した場合、DS_OK が返され、lpListener は、リスナーの現在の速度を返す。

IDirectSound3DListener::GetAllParameters

IDirectSound3DListener::SetDistanceFactor

HRESULT SetDistanceFactor(
D3DVALUE flDistanceFactor, DWORD dwApply);

の

- 成功した場合、DS_OK が返され、flDistanceFactor は、リスナーの現在の速度を返す。

flDistanceFactor

、成功した場合、DS_OK が返され、lpListener は、リスナーの現在の速度を返す。

dwApply

、成功した場合、DS_OK が返され、lpListener は、リスナーの現在の速度を返す。

DS3D_DEFERRED

ffff fff, IDirectSound3DListener::CommitDeferredSettings ffff, 成功した場合、DS_OK が返され、lpListener は、リスナーの現在の速度を返す。

IDirectSound3DListener::GetDistanceFactor

dwApply
, ,,,,fffff,Ž, ’ ~%oo, ’ Ž,,,

```
ffff fff, IDirectSound3DListener::Com
mitDeferredSettingsffff, E, ,, ', ,,,
• , ', • ,,, “, ĆŽ, Ž ,,,, %” ,,,
```

’, ‘Ž, ,, ffff,,,,3Dffff ffff,‘,,3D •
, ŒŽ, ,

 $(0, 1.0, 0)$

```
ffff fff, IDirectSound3DListener::Com
mitDeferredSettingsffff, E, ,, ', ,,,
• , ', • ,,, " , EŽ, Ž ,,,,,,
```

’, ‘Ž, , ffff,,,,,3Dffff ffff, ‘, 3D •
, ŒŽ,,

```
ffff fff,IdirectSound3Dlister::Comm
tDeferredSettingsffff,Œ, ,, ', ,,, •
, ',• ,,, “, ŒŽ,Ž ,%”,,,,

```

’, ‘Ž, „, ffff,,,,,3Dffff ffff,‘,,3D •

, $\mathbb{E}\check{Z}$, ,

DS3D_DEFAULTROLLOFFFACTOR (1.0)

IDirectSound3DListener::GetRolloffFactor

IDirectSound3DListener::SetVelocity

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

の

- $\mathbb{E},,,\text{DS_OK}$ \check{Z} ”,,,DSERR_INVALIDPARAMfff f f,•,

x, y, z

D3DVALUEfff,’ fff , <“ , •,Ž,,,

dwApply

’, ,,,fffff,Ž,’ ~%o,’Ž,,,

DS3D_DEFERRED

ffff fff,IDirectSound3DListener::Com
mitDeferredSettingsffff,Ⓔ, ,,’ , ,,,
• , ,’• ,,, ’“ , $\mathbb{E}\check{Z}$,Ž ,%o”,,,

DS3D_IMMEDIATE

’,Ž, , , ffff,,,,,3Dffff ffff,’ ,3D •
, $\mathbb{E}\check{Z}$, ,

のの **IDirectSound3DListener::SetPosition**

(0,0,0)

IDirectSound3DListener::GetVelocity

IDirectSoundBuffer

DirectSoundBuffer

IDirectSoundBuffer のの

•

GetCaps

GetFormat

GetStatus

SetFormat

ffffŠ—

Initialize

Restore

Š—

GetCurrentPosition

Lock

Play

	SetCurrentPosition
	Stop
	Unlock
ffffŠ—	GetFrequency
	GetPan
	GetVolume
	SetFrequency
	SetPan
	SetVolume
の COM	IDirectSound3DBuffer
IUnknown	
の	
AddRef	
QueryInterface	
Release	

IDirectSoundBuffer::GetCaps

HRESULT GetCaps(LPDSBCAPS lpDSBufferCaps);

DirectSoundBuffer

- Ɛ,,,DS_OK Ž”,,,DSERR_INVALIDPARAMff f f,•,

lpDSBufferCaps

DSBCAPS “,,ffff ffff ffff, ”Ž,,,,

DSBCAPS **IDirectSound::CreateSoundBuffer**

DSBUFFERDESC

ののののののの

DSBCAPS **dwFlags** **DSBUFFERDESC**

の **DSBCAPS**
DSBCAPS_LOCHARDWARE DSBCAPS_LOCSOFTWARE の
の

DSBUFFERDESC

DSBCAPS, DSBUFFERDESC, IDirectSoundBuffer,
IDirectSound::CreateSoundBuffer

IDirectSoundBuffer::GetCurrentPosition

HRESULT GetCurrentPosition(LPDWORD lpdwCurrentPlayCursor,
LPDWORD lpdwCurrentWriteCursor);

- `DS_OK` `DSERR_INVALIDPARAM`
`DSERR_PRIOLEVELNEEDED`

lpdwCurrentPlayCursor

DirectSoundBufferfffff“,`DSERR_INVALIDPARAM`,`DSERR_PRIOLEVELNEEDED`,fffff“,fffff
,, fff,Ž’,,,

lpdwCurrentWriteCursor

DirectSoundBufferfffff“,`DSERR_INVALIDPARAM`,`DSERR_PRIOLEVELNEEDED`,fffff“,fffff
,, fff,Ž’,,,

15 の

lpdwCurrentPlayCursor の

IDirectSoundBuffer, **IDirectSoundBuffer::SetCurrentPosition**

IDirectSoundBuffer::GetFormat

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

- `DS_OK` `DSERR_INVALIDPARAM` f f f , ,

lpwfxFormat

ffff“,ffff f f,,, ,Š,WAVEFORMATEX “,ffff `DSERR_INVALIDPARAM`,—
,ffff fff,Ž“,,, ,NULL,Ž’,,,

dwSizeAllocated

WAVEFORMATEX “,fff^,fff DirectSound,ffff, ,,,,,,dwSizeAllo
cated,fff ,,,, ,WAVEFORMATEX “,,,,‘,fff,—‘,, , ,,,,

lpdwSizeWritten

WAVEFORMATEX “, ,,,,fff ,Ž,• ,ffff ,fff f,NULL,,,

WAVEFORMATEX

の DirectSoundBuffer の
lpdwSizeWritten の

IDirectSoundBuffer, **IDirectSoundBuffer::SetFormat**

IDirectSoundBuffer::GetFrequency

`HRESULT GetFrequency(LPDWORD lpdwFrequency);`

の 1 の

- `DS_OK` `DSERR_INVALIDPARAM` f f f , ,

DSERR_CONTROLUNAVAIL
DSERR_INVALIDPARAM
DSERR_PRIOLEVELNEEDED

lpdwFrequency
f fff ffff, ,,,, ,Ž” ,Ž,• ,ffff
の 100 100,000 の

IDirectSoundBuffer, **IDirectSoundBuffer::SetFrequency**

IDirectSoundBuffer::GetPan

HRESULT GetPan(LPLONG lplPan);

- Ǝ,,,DS_OK Ž”,,,Ž,ff f f,•,
DSERR_CONTROLUNAVAIL
DSERR_INVALIDPARAM
DSERR_PRIOLEVELNEEDED

lplPan
‰, ,“ffff,Ž,• ,ffff
-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);

∞

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

lpdwStatus
ffff ffff, ‘,Ž,• ,ffff ‘,Ž,’ ,’,,
DSBSTATUS_BUFFERLOST
ffff,Ž,,,, ,,,fff, ,‘ffff,,—,,

DSBSTATUS_LOOPING

ffff,f f,,, ',-
',,, ffff,ffff f, Ɛ, ,Ž,, ,, , Ž,, ffff, , ,,,,,,,'^,,

DSBSTATUS_PLAYING

ffff, ',, ,, , ',,,,,ffff,Ž,,,
IDirectSoundBuffer

IDirectSoundBuffer::GetVolume

HRESULT GetVolume(LPLONG lplVolume);

∞∞

- Ɛ,,,DS_OK Ž”,,,Ž,ff f f,•,

DSERR_CONTROLUNAVAIL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

lplVolume

Ž’,,,DirectSoundffff,fff f,Ž,• ,,ffff

100dB の 0 -10,000 の 0 ∞∞

-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**ff f f,•,

lpDirectSound

DirectSoundBufferffffff,Ɛ ,,DirectSoundffffff,,fff

lpDSBufferDesc

ffff ffff, Š‰,Ž—,,',Ž,,,DSBUFFERDESC “,,ffff

IDirectSound::CreateSoundBuffer

IDirectSoundBuffer::Initialize

DirectSound の

DSBUFFERDESC, **IDirectSound::CreateSoundBuffer**,

IDirectSoundBuffer

IDirectSoundBuffer::Lock

```
HRESULT Lock(DWORD dwWriteCursor, DWORD dwWriteBytes,
             LPVOID lpIPvAudioPtr1, LPDWORD lpdwAudioBytes1,
             LPVOID lpIPvAudioPtr2, LPDWORD lpdwAudioBytes2,
             DWORD dwFlags);
```

の

- E,,DS_OK Ž”,,,Ž,ff f f,•,

DSERR_BUFFERLOST

DSERR_INVALIDCALL

DSERR_INVALIDPARAM

DSERR_PRIOLEVELNEEDED

dwWriteCursor

ffff,ŠŽ^,,fff,ŠŽ,,^,,, fff^,fffff

*dwFlags*fff f,DSBLOCK_FROMWRITECURSORfff,Ž’,,,,, ,,,fff
f,-Ž,,,

dwWriteBytes

fff,,ffff—^, fff^,fff ffff ffff,Š”“,,%o,,,,, ffffffff ,,^,,,

lpIPvAudioPtr1

fff,,ffff, ,ffff,,ffff

lpdwAudioBytes1

lpIPvAudioPtr1fff f,Ž,,fff ,Ž,• ,ffff ,’,dwWriteBytesfff f,, ,,
lpIPvAudioPtr2,ffffff f,Ž,ffff,Ž,

lpIPvAudioPtr2

fff,,ffff ffff,2”—,—^,ffff ,fff f,’NULL,
lpIPvAudioPtr1fff f,ffff ffff,fff,,—^“,Ž,

lpdwAudioBytes2

lpIPvAudioPtr2 fff f,ffff,,fff ,Ž,• ,ffff
lpIPvAudioPtr2,NULL, ,’,0,,,

dwFlags

fff,ffff, ,fff ^%o,’Ž,,,

DSBLOCK_FROMWRITECURSOR

E , ,,,f ff,,fff, ,

,,,,IDirectSoundBuffer::GetCurrentPosition,E, ,•—

,,, ,fff,Ž’,,,dwWriteCursorfff f,-Ž,, ,fff,”^Ž’,,,

のののののの *lpIPvAudioBytes2* NULL のの

lpIPvAudioPtr2

lpdwAudioBytes2

NULL

DirectSound の

IDirectSoundBuffer::Lock

DirectSound

IDirectSoundBuffer::Unlock

Warning

DirectSoundBuffer ffffff f,“, , , , ,DirectSoundBuffer ffffff f,—E,ffff f f,Š,,,, ,f f,— E,,, ,fff,fff f fff,”,,,,, ffff,fff ffffffff“,ffff ffff,ff ff,,, IDirectSoundBuffer::Unlock,E, , , ,ffff ffff,fff f fff,“,, ,

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

IDirectSoundBuffer::Play

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

の

- E,,,DS_OK Ž”,,,Ž,ff f f,•, **DSERR_BUFFERLOST** **DSERR_INVALIDCALL** **DSERR_INVALIDPARAM** **DSERR_PRIOLEVELNEEDED**

dwReserved1

fff f,— , , , ,0,,,,,,

dwReserved2

fff f,— , , , ,0,,,,,,

dwFlags

ffff, •,Ž’, ^%,’,Ž,,

DSBPLAY_LOOPING

f fff ffff, —’,“,,, ffff,ŠŽ~, , , , ’Ž,— Ž,,,E,•, ,fff,ffffff ffff ffff, ’, ’,•,—,,

の

□□□□□□□□

DSBPLAY_LOOPING の

の

Note `,,,ffff ffff,,,, ffff fff,,,ffff,Œ, ,,‘,IDirectSound::SetCooperativeLevelffff,Œ, , <’fff,Ž’,,•—`
`,,, ’ ,DSSCL_NORMAL,Ž’,, ,,IDirectSound::SetCooperativeLevelffff,`
`Œ, ,,,,,,`
IDirectSoundBuffer::Play`ffff,DSERR_PRIOLEVELNEEDEDff ’,•,`

IDirectSoundBuffer, **IDirectSound::SetCooperativeLevel**

IDirectSoundBuffer::Restore

`HRESULT Restore();`

の **IDirectSoundBuffer**

- `Œ,,,DS_OK Ž”,,,Ž,ff f f,•,`
DSERR_BUFFERLOST
DSERR_INVALIDCALL
DSERR_INVALIDPARAM
DSERR_PRIOLEVELNEEDED

IDirectSoundBuffer::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,ff f f,•,`
DSERR_INVALIDCALL
DSERR_INVALIDPARAM
DSERR_PRIOLEVELNEEDED

dwNewPosition

`ffff ffff, ,Ž—,,,fff^,fffff’`

の

,0, Ž” ,Œ ,ffffŒŽ,ffff,,, ,ŒŽ, IDirectSound::CreateSoundBu
fferffff,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,ff f f,•,
DSERR_CONTROLUNAVAIL
DSERR_GENERIC
DSERR_INVALIDPARAM
DSERR_PRIOLEVELNEEDED

lPan

%o,fffff,“fff f,Ž“,, ,,’”^,-10,000,, 10,000,,, 100dB,”^,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,ff f f,•,
DSERR_CONTROLUNAVAIL
DSERR_GENERIC
DSERR_INVALIDPARAM
DSERR_PRIOLEVELNEEDED

lVolume

ffff ffff,—<,,, <,fff f ’”^,0(0dB(dB) fff f• ,,) ,,-10,000 -100Db Ž ,-%o ,,,, DirectSound,Œ ,•,ff f,,,,,

の 100dB 0 のの

10dB の DirectSound 20dB 4 の 1

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

IDirectSoundBuffer::Stop

HRESULT Stop();

の

- DS_OK, DSERR_INVALIDPARAM, DSERR_PRIOLEVELNEEDED

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

IDirectSoundBuffer::Stop

IDirectSoundBuffer::Play の

の dB

IDirectSoundBuffer, **IDirectSoundBuffer::Play**

IDirectSoundBuffer::Unlock

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

の

- DS_OK, DSERR_INVALIDCALL, DSERR_INVALIDPARAM, DSERR_PRIOLEVELNEEDED

lpvAudioPtr1
IDirectSoundBuffer::Lock ffff, lpvAudioPtr1 ffff, f, ffff

```

dwAudioBytes1
    Ž ,lpvAudioPtr1fff f, ,,,,fff IDirectSoundBuffer::Lockffff,,,•,,
    ,fff ,%o,,,,,,,,
lpvAudioPtr2
    IDirectSoundBuffer::Lockffff, lplpvAudioPtr2 fff f,Ž“,,,• ,ffff
dwAudioBytes2
    Ž ,lpvAudioPtr2
    fff f, ,,,,fff IDirectSoundBuffer::Lockffff,,,•,,,fff ,%o
    ,,,,,,
    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

,, “,fff”^,fff

vPosition

,,ffff ffff,☉ ,^,< ,, D3DVECTOR “

vVelocity

,,ffff ffff,☉ ,“,< ,, D3DVECTOR “

dwInsideConeAngle

“ffff ffffffff f f,Š“

dwOutsideConeAngle

Š‘ffff ffffffff f f,Š“

vConeOrientation

,,,ffffffffff ffffffff f f,☉ ,•☉,☐ ,, D3DVECTOR “

lConeOutsideVolume

Š•fff f,f f

flMinDistance

☐—

flMaxDistance

‘☐—

dwMode

,,ffff —f f Ž,’Ž,,

DS3DMODE_DISABLE

3Dffff —,—☉,,, ffff,fff ‘,’ ,,•,,,,,Š,,

DS3DMODE_HEADRELATIVE

ffffffffff f(^ “ •☉)

,fff ,fff f,‘Š☉,,, ,,f f,, ffff, ‘fff f,fff fff f,• ,,,,,,
Ž““,, ,,,, “fff f,•,,,,,,,

DS3DMODE_NORMAL

• , — ,,,ffffff f,,,,,

DS3DLISTENER

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

○○○

IDirectSound3DListener::GetAllParameters

IDirectSound3DListener::SetAllParameters

dwSize

,, “,fff”^,fff

vPosition, vVelocity, vOrientFront, vOrientTop
fff ,^ “ ‘•☉,,, •☉,,,,☐ ,, D3DVECTOR “

flDistanceFactor, flRolloffFactor, flDopplerFactor
,,,☉ ,☐— f fff,,ffff —‘

DSBCAPS

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

IDirectSoundBuffer::GetCaps

DirectSound

の

dwSize

,, “,fff”^,fff

dwFlags

ffff fffffff”—,Ž’,,fff

DSBCAPS_CTRL3D

ffff,,, Ć,Ž—,,ffffff ffff,,,

DSBCAPS_CTRLFREQUENCY

ffff,Ž” Ć”—,Ž,,,,,,,,,

DSBCAPS_CTRLPAN

ffff,ff Ć”—,Ž,,,,,,,,,

DSBCAPS_CTRLVOLUME

ffff,fff f Ć”—,Ž,,,,,,,,,

DSBCAPS_GETCURRENTPOSITION2

IDirectSoundBuffer::GetCurrentPosition , f ff, ,, “ ,Ž—
,,,,,,,,,Ž, DirectX ,,DirectSound
,, ,,,f ff,’Ć,,, f ff,ffff f f,ffff f,,%o, Ž ,%o,,, —
,, ,,, Ć ,, DSBCAPS_GETCURRENTPOSITION2fff,Ž’,,,, ffff
fff,,,, Š, ^,Ž“,%o”,,,
,,fff,Ž’,,,, ‘,,,%o<“ ,ĆŠ ,,,^Ž,,,, ,fff,%o<—
,,,ffff f f ffff f,,,,,,,,,’^,,, ,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,Direc
tSound ,Ž, DirectSound ,Ž,,,,ffff fff,,ffff, ‘,,,,,
,, ff ff, DSSCL_EXCLUSIVE
,,, DSSCL_WRITEPRIMARYfff,‘fff,Ž,DirectSound
ffff fff, ‘,, ,—Š,, , ,’ffff fff,,ff ff,%o,•,,,,,

DSBCAPS_LOCHARDWARE

DSBCAPS_STATIC ,Ž’,,,,,, ffff,< “,f ffff fffff,Ž—,,
,,ffff,f ffff fffff,ff f,,,,, ,,,•—,,f ffff fff,—
•%o”, , **IDirectSound::CreateSoundBuffer** ,Ć, ,Ž”,,
ffff fff,ffffff ffff,,ffff,,,—%o”,,,,,• ,•—,,

DSBCAPS_LOCSOFTWARE

DSBCAPS_STATIC ,Ž’,,f ffff ff f,—

```

        %",,,,,, ffff,< " ,ffffff fff,Š",, fffffff fffff,Ž—,,
DSBCAPS_PRIMARYBUFFER

        ffff,ffffff ffff ffff,,,,*,
        ,,,',Ž',,,,,, fffff ffff ffff, ,,,
DSBCAPS_STATIC

        ffff,ffffff ffff f f,,,Ž',,,,,*,
        ^",,,,,ffff,"f f,,, ,,%o", ,,, ,,,,ffff,f ffff fff,—<,
DSBCAPS_STICKYFOCUS

        ffff ffff,ff ff," ,*,
        ,fff, IDirectSound::CreateSoundBuffer,Œ, ,Ž',,,,,,
        ,fff,ffff,,,, ,f f,'DirectSound ,Ž',,,,ffff fff, ,' ,,,
        DirectSound ,Ž',,,,ffff fff,,,ffffff ff ff ffff, ,' ,,,,,,
        ,,'ffff fff,* ffff, %o,,, fffff ff ff ffff,—
        ,' ,,,,f f,f f
        ,,,,ffff,fff,Š,fffffffff,*,Ž f f~Š,ffff fff ,,,,%o%o
        (ActiveMovie™),,—, ,,, f f,,,', DirectSound
        ffff fff, ,' ,,,, * ,ffffff ff ff—•,ffff ffff,,, %o,,,
dwBufferBytes
        ,ffff,fff^,fff
dwUnlockTransferRate
        IDirectSoundBuffer::Unlock,Œ, ,,,,ffff fff,f f," ,,,f f, ,•—
        ,kfff,Ž',, fffff fff,ffff fff, IDirectSoundBuffer::Unlock,Ž ,,,,
        —,ŽŠ,Œ',,,, ,',Ž',,,,,, fffffff," ,,,,ffffff ffff,,,f f, —•
        —," , ,
        f ffff ffff,,,f f, ffff,ffff f f,f f,,,%o" ,,,, ,',,,,%o" ,,, ,
        ,,""" ,Œ',,,, ,,,
dwPlayCpuOverhead
        ,ffff ffff,ffffff,*,—,f f fff,Ž, —ffff,f fff f,Ž',,
        f ffff ffff,,,, ,fff,ffffff,ffff ffff,,, ,,,, ,,,
        fffffff ffff,,, ,fff,ffff ff fff,ffff fffff," ,^',,
DSBCAPS IDirectSound::CreateSoundBuffer
DSBUFFERDESC

```

⑦

DSBCAPS	dwFlags	DSBUFFERDESC
		⑦ DSBCAPS
DSBCAPS_LOCHARDWARE		DSBCAPS_LOC SOFTWARE
	⑦	DSBUFFERDESC
∞∞∞		

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^,fff

dwFlags

„, DirectSoundBuffer fffffff, „„,Š,—,Ž•,, ^%o„„„„„• ,Ž’ „„,

DSBCAPS_CTRL3D

ffff„„, Ć,Ž,ffffff ffff„„,

DSBCAPS_CTRLALL

ffff„„„„ Ć”—,Ž„„„„„„,

DSBCAPS_CTRLDEFAULT

ffff,ffffff,ffff f fffff,Ž„„„„ „„,DSBCAPS_CTRLPAN
DSBCAPS_CTRLVOLUME„„,DSBCAPS_CTRLFREQUENCYfff,Ž’ „„
„„ „„„

DSBCAPS_CTRLFREQUENCY

ffff,Ž” Ć”—,Ž„„„„„„,

DSBCAPS_CTRLPAN

ffff,ff Ć”—,Ž„„„„„„,

DSBCAPS_CTRLVOLUME

ffff,fff f Ć”—,Ž„„„„„„,

DSBCAPS_GETCURRENTPOSITION2

IdirectSoundBuffer::GetCurrentPosition , f ff, „„, ’,Ž—
„„„„„„,Ž, DirectX „,DirectSound
„ „„,f ff,’Ć„„„ f ff,ffff f f,ffff f„„,‰, Ž ,‰„„, —
„ „„„ Ć „„,DSBCAPS_GETCURRENTPOSITION2fff,Ž’ „„, ffff
fff„„„, Š, ^,Ž“„‰„„„
„,fff,Ž’ „„„, ‘„„,‰„„,“ „,ĆŠ „„„,~Ž„„„ „,fff,‰„„,—
„„,ffff f f ffff f„„„„„„,^„„„ „,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,Direct
tSound ,Ž, DirectSound ,Ž„„,ffff fff„,ffff, ‘„„„„„„
„„ ff ff, DSSCL_EXCLUSIVE
„„„ DSSCL_WRITEPRIMARYfff,‘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** ,Ć, „,Ž”„„
ffff fff,ffffff ffff„„,ffff„„„,—‰„„„„„„,• „„,•—„„

DSBCAPS_LOCSOFTWARE

DSBCAPS PRIMARYBUFFER**DSBCAPS_STATIC**

DSBCAPS STICKYFOCUS

dwBufferBytes

.,ffff,fff'^,fff .,,',ffffff ffff, ,,,,,,,,,,,,,,

dwReserved

„fff,—,,,, ž—,,,,

lpwfxFormat

```
ffff,,waveff fff,Ž',,, “,,ffff ,',ffffff ffff,,NULL,,,,,,
ffff fff,ffffff ffff,ff fff,fff,,,, IDirectSoundBuffer::SetFormat
,Ž,,,,,
```

dwFlags DSBCAPS LOCHARDWARE

DSBCAPS LOC SOFTWARE

DSBCAPS LOCHARDWARE

DSBCAPS LOC SOFTWARE

④ **DSBCAPS** **dwFlags**
DirectSoundBuffer

dwBufferBytesDirectSound 

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

,, “,fff”^,fff

dwFlags

ffff,”—,Ž’,, ^%o,,,,,,,,,• ,Ž’,,,

DSCAPS_CERTIFIED

,,ffff,Microsoft,,,ŽĚ,,• ,,,,

DSCAPS_CONTINUOUSRATE

ffff,dwMinSecondarySampleRate ,dwMaxSecondarySampleRate
 fff’,Š,,,,,ffffffff f,ff f,, ^”,,,Ž , —f f,—(,,Ž” ,
 ,,fff Hz ’,,,,,^—,,

DSCAPS_EMULDRIVER

ffff,,ffff f,,,DirectSound
 ffff,,, ,,ff fff f f fff<”’,,ffff f,,,, Ž ”—,%o,—‘,,,

DSCAPS_PRIMARY16BIT

ffff,,ffffffffffff,fffff ffff ffff,ff f,,

DSCAPS_PRIMARY8BIT

ffff,,ffffffffffff,fffff ffff,ff f,,

DSCAPS_PRIMARYMONO

ffff,%offff ffff,ff f,,

DSCAPS_PRIMARYSTEREO

ffff,ffffffffffff ffff,ff f,,

DSCAPS_SECONDARY16BIT

ffff,16ffffffffffff,f ffff fffff fffff ffff,ff f,,

DSCAPS_SECONDARY8BIT

```

ffff,,ffffffff,f ffff fffff fffff ffff,ff f,,
DSCAPS_SECONDARYMONO
ffff,f ffff fffff'%'ffffff ffff,ff f,,
DSCAPS_SECONDARYSTEREO
ffff,f ffff fffff ffff fffff ffff,ff f,,

dwMinSecondarySampleRate      dwMaxSecondarySampleRate
,,ffff,f ffff fffff ffff,ff f,, ' ,ffffff f,Ž'

dwPrimaryBuffers
ff f,,ffffff ffff, %0,ff f,,,,,

dwMaxHwMixingAllBuffers
f ffff,ffffff,,ffff, Ć, ,Ž',, ,,fff, dwMaxHwMixingStaticBuffers
, dwMaxHwMixingStreamingBuffers , Ć,, ,,
ff f,ff fff,,," ,,

dwMaxHwMixingStaticBuffers
ffffff ffff ffff, ' ,Ž',,

dwMaxHwMixingStreamingBuffers
fff f ffff ffff, ' ,Ž',,

dwFreeHwMixingAllBuffers, dwFreeHwMixingStaticBuffers,
dwFreeHwMixingStreamingBuffers
Š,“,,,,,ffff, f ffff fffff”—, '
ffff fff, f ffff ff f,ffffff ffff ffff,Š,“,,,,,—
%”,,,,,”,,,, ,,,',Ž,,,,, “,,,,,’ ,‘fffff”—
,Ž',,fff,”Š,, ,Š,“,,,,,ff f,“,,,,,

dwMaxHw3DAllBuffers, dwMaxHw3DStaticBuffers,
dwMaxHw3DStreamingBuffers
ffff,f ffff,,^”—, < ,,,,,, ,ff f,,,,,

dwFreeHw3DAllBuffers, dwFreeHw3DStaticBuffers,
dwFreeHw3DStreamingBuffers
Š,“,,,,,ffff, f ffff,,^Ć,—, < ,,,,,, ,ff f,,,,,

dwTotalHwMemBytes
ffffff ffff ffff,Š”,,ffff f f ,fff—,fff^,fff

dwFreeHwMemBytes
ffff f f ,ff fff,fff^,fff

dwMaxContigFreeHwMemBytes
ffff f f ,ff fff, ,',~'ffff,fff'^,fff

dwUnlockTransferRateHwBuffers
f ffffffff ffff ffff fff ffffffff,”,,,, ,f f,“,,,f f,,•—,k
fff,< , ,f f,fff,“—,
IDirectSoundBuffer::UnlockffffĆ, ,Ć'ŽŠ,Ć',,

dwPlayCpuOverheadSwBuffers
ffffff ffff ffffffff,”,,,, ,fffff,,,,,•—, CPU,f fff f, —
f f fff,' ,,,ffffff ffffffff,,fffff““^,•%0,,

```

```

fff,,,,,ffffff ffff,“f f, f f,,,,,““,•—,,,,,, “—
,f fffffff, f f fff, fffff,ffff ffff,,,, —,,,,,,
dwReserved1      dwReserved2
,,,fff,—,,,, Ž—,,,
IDirectSound::GetCaps

```

●’

```

IDirectSoundBuffer      のの      IDirectSound
ののの

```

DS_OK

```
—<, Ą,Š—,,
```

DSERR_ALLOCATED

```
—<— fff,,,,ff f,‘Ą, ,,,,,,Ž—’,,,,,Ž”,,
```

DSERR_ALREADYINITIALIZED

```
ffffff,,, Š%o,,,,
```

DSERR_BADFORMAT

```
Ž’,,,wave,ff f,,,
```

DSERR_BUFFERLOST

```
ffffff,Ž,,,, ffff,,,,,,
```

DSERR_CONTROLUNAVAIL

```
Ą, ,,,,—<, Ą fff f ff,, ,—%o”,,,
```

DSERR_GENERIC

```
–Š’,ff , DirectSound fffffff,’<,,,
```

DSERR_INVALIDCALL

```
,,Š ,,ffffff,Ą , ‘,,—Ą,,,
```

DSERR_INVALIDPARAM

```
–Ą,fff f,Š “,,,
```

DSERR_NOAGGREGATION

```
ffffff,COM %o,ff f,,,
```

DSERR_NODRIVER

```
Ž—,,ffff ffff,—,,,,
```

DSERR_OTHERAPPHASPRIO

```
,,’%o<,,,,,Ž—,,,,
```

DSERR_OUTOFMEMORY

```
DirectSound fffffff,Ą, ,,—<Š—,,,, ,fff,Š,“ ,,,,,,,
```

DSERR_PRIOLEVELNEEDED

```
Ą, ,Š , Ą,,,,,•—,— fff,Ž,,,,
```

DSERR_UNINITIALIZED

```
‘,ffff,Ą, ,,’IDirectSound::Initializeffff,Ą, ,,,,, ,,,Ą, , Ą,,,
```

”
DSERR_UNSUPPORTED
E, ,Š ,,,,ff f,,,,,