

**Microsoft® DirectX™ 3**  
*ffffffŠ”fff*



# 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

## DirectSound *f* *fffff*

のDirectSound のの DirectXのの

- *f fffff*—
- *ffffff fff*
- *ffffff* *ffff* *fff*
- *ffff* *ffff*
- *'fff*
- *ffff,*“

## *f* *fffff*—

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1 の

の

**ffffff** □ **fff**

の DirectSound DirectSound *IDirectSound*  
(COM) の の の

の DirectSound

DirectSound

*IDirectSoundBuffer*

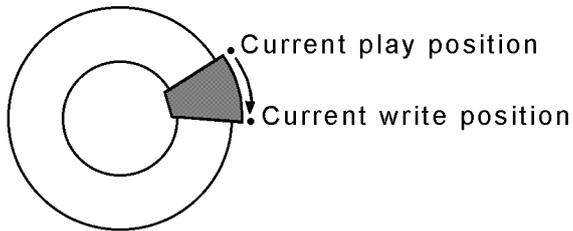
の の の

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* fffffff
- *DirectSoundBuffer* fffffff

**DirectSound** fffffff

の

*IDirectSound*

**DirectSoundCreate**

**IDirectSound**

DirectSound

DirectSound

**DirectSoundEnumerate**

Windows

の

DirectSound

**Note** DirectSound, ffffffff, IDirectSound, ffff, IDirectSoundBuffer ffff ffff  
, ' < , , , Cœœ, fff' < , Š , , ,

## DirectSoundBuffer ffffffff

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

の DirectSound

### IDirectSoundBuffer::Lock

**IDirectSoundBuffer::Play** DirectSound  
 の

### IDirectSoundBuffer::Play の

の  
 の

DirectSoundBuffer  
**IDirectSound::CreateSoundBuffer** DSBCAPS\_STATIC  
 ( の )

PCI の  
 の

## fffffff □ ffff □ fff

DirectSound  
**IDirectSound::GetCaps**  
 DirectSound

## ffff □ ffff

DirectSound  
 DirectSound HAL Windows の  
 HAL の  
 Windows の DirectSound HAL

- f fff < Š , , , ffff f, Ž “ , % •

- *f fff*<Š, ”\*%o
- *f ffff,—E, ,“,’ ,Ž*
- *f ffff,—E, ,‘ —<,’*

DirectSound の DirectSound の  
Windows NT® Windows 3.1 の DirectSound の

Windows 95

**’fff**

DirectSound

4

**IDirectSound::SetCooperativeLevel**

**IDirectSoundBuffer::SetFormat**

**IDirectSound::Compact**

∞

DirectSound

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

DirectSound

DirectSound

wave

wave

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

**IDirectSoundBuffer::Lock**

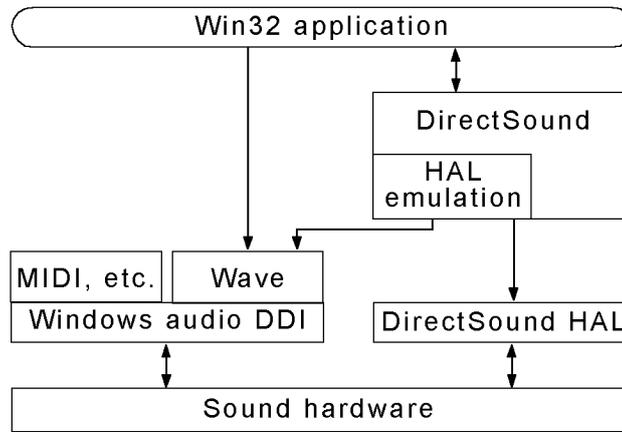
DirectSound

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

*IDirectSound*

**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*
- *fffff ffff,, ,,ffff*

### ***fffff***

DirectSound の  
DirectSound ののの DirectSound  
の

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

ののののの DirectSound

### ***f ffff fffff fff***

DirectSound の

のの

DirectSound

のの

ののののの

### ***fffff ffff,, ,,ffff***

DirectSound □□□□□□□□□□ □□□

□□□□□□□□□□

□□□

□□□

□□□

DirectSound の  
(DirectSound

の □□□□□□□□□□  
の□□□ ののののののののの

### 3Dffff

DirectSound

*IDirectSound3DBuffer*

*IDirectSound3DListener*

の  
の

3D の

- %' ^, "Ž
- fff
- ffff f f
- '←, '←
- ^, “
- Direct3D, “
- ’’^, ←—‘
- ffff%CE, ffff%CE

### %' ^, "Ž

ののの

- fff f %o, ,', fff ,—,,, ,%o, ,,,,,, ,,CE ,f fff,CE,,,,,
- “,,, fff ,%o•CE,,”,,,%o, fff ,%oŽ,“,,, Ž,,,,, , ,,, (,,,,Ž•ŽŠ, - 1ff•,,)
- %o fff ,CE,,,,,%o, ‘,,,%o,”,,,,, %o ,,,, ,,,, ,Ž•CEŠ”ŠŽ,,, ,,, fff ,%o,,%o,,, , Ž,“%o,“,,,,, %o ,,,

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

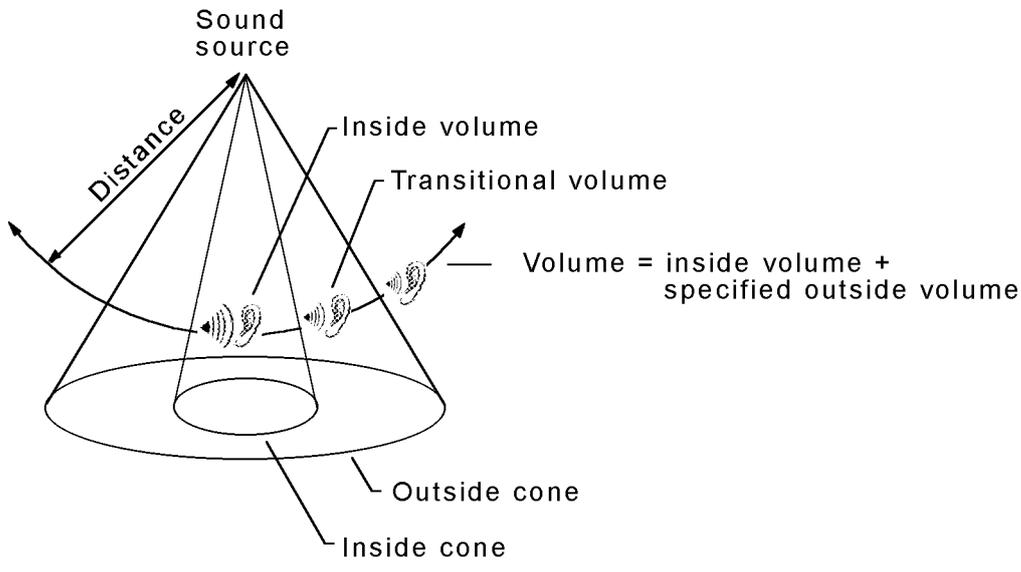
□□□□□□□□

DirectSound の□□□□□□□□

のの(DirectSound  
の)ののの-10,000 の

0 ののの

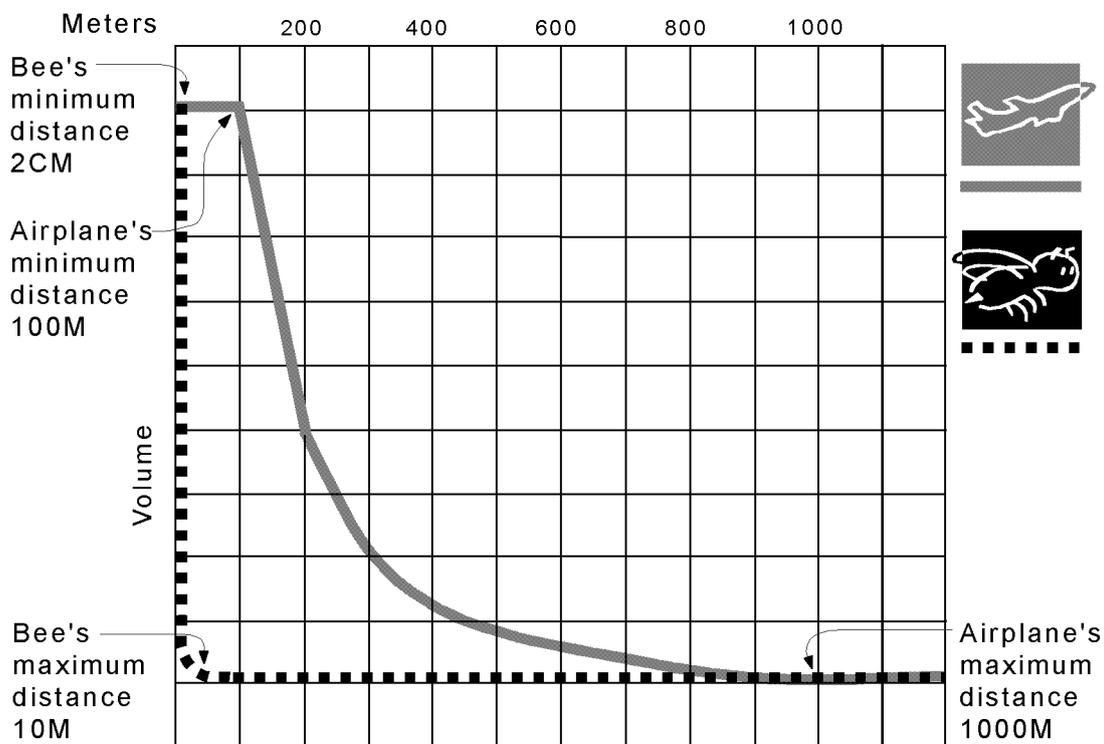
□



```

IDirectSound3Dbuffer
□□□□□□□□ □□□□□□□□□□□□ 0
○○
□'←, □'←
(0)○○○○○
(16)○○○○ 100 ○
2 ○
200 4
○○○

```



~, “  
 の 3D                      3D ののの3D                      3D のの 3D  
 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 ののののの 0.30480006096 1

**ffff%oœ,ffff%oœ**  
3D のの

DirectSound の 3D の

## DirectSound **ffff** □ **ffffŠ—**

のの DirectSound

- *IDirectSoundfff ffff*
- *IDirectSound3DBuffer fff ffff*
- *IDirectSound3DListener fff ffff*
- *IDirectSoundBuffer fff ffff*

## **IDirectSoundfff** □ **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**

∞  
DSSCL\_NORMAL ∞

**ff** □ **f,Š** < □ '

*IDirectSound*  
∞ **IDirectSound::GetSpeakerConfig**  
**IDirectSound::SetSpeakerConfig**

**f** □ **fffffffŠ** —

**IDirectSound::Compact**  
∞

**IDirectSound3DBufferfff** □ **ffff**

*IDirectSound3DBuffer* ∞ 3D ∞

∞ **IDirectSound3DBuffer** ∞ ∞ ∞

- *IDirectSound3DBufferfff* ffffffff,Ž“
- fffffff f‘
- '←', '←'
- ' f f
- ^,“
- ffff ffffffff f f

**IDirectSound3DBufferfff** □ **fffffff,Ž**“

*IDirectSound3DBuffer* ∞ 3D  
**IDirectSound::CreateSoundBuffer** **DSBUFFERDESC**  
**dwFlags** ∞ **DSBCAPS\_CTRL3D**  
**IDirectSoundBuffer::QueryInterface**  
*IDirectSound3DListener* ∞

```
// DSBCAPS_CTRL3D, Ž-, , lpDsbSecondary, □□, ,
hr = lpDsbSecondary->QueryInterface(IID_IDirectSound3DBuffer,
    &lpDs3dBuffer);
if (SUCCEEDED(hr)) {
    // , , ffff, 3Dfff□f, □' , ,
    .
    .
    .
}
```

DirectSound 3D ∞  
3D ∞  
3D

---

**Note** *ff*  $\mathbb{E}$ ,3D  $\text{---}$ , $\leftarrow$  ,, ,,,,  
DSBCAPS\_CTRL3D,DSBCAPS\_CTRLPAN,—  
•,Ž’ ,,,,,,DirectSound,— $\leftarrow$  , Ž” ,,

---

***fffff***  $\square$  ***f***‘  $\square$   
3D の

の  
*IDirectSound3DBuffer*  
の

**IDirectSound3DBuffer::GetAllParameters**

$\square$  ‘ $\leftarrow$ ’,  $\square$  ‘ $\leftarrow$ ’  
3D

の

の **IDirectSound3DBuffer::SetMinDistance**

**IDirectSound3DBuffer::GetMinDistance**

の **IDirectSound3DBuffer::SetMaxDistance**

**IDirectSound3DBuffer::GetMaxDistance**

‘ $\square$  ***f***  $\square$  ***f***’

( ) の の の

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

^’ ‘‘  
,

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

DirectSound  
**IDirectSound3DBuffer::SetVelocity**  
**IDirectSound3DBuffer::GetVelocity**  
の の の

***ffff***  $\square$  ***ffffffffff***  $\square$  ***f***  $\square$  ***f***  
3D  
*IDirectSound3DBuffer*  
の

**IDirectSound3DBuffer**



の dwFlags DSBCAPS\_CTRL3D  
**IDirectSoundBuffer::QueryInterface**  
**IDirectSound3DListener** の

```
// DSBCAPS_CTRL3D, 3D, lpDsbPrimary, , ,
hr = lpDsbPrimary->QueryInterface(IID_IDirectSound3DListener,
    &lpDs3dListener);

if (SUCCEEDED(hr)) {
    // 3D, 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, 3D, , , , , , fff, 3D  
 , fff “,(1,2,3), ’ , , , , fff, 3D, fff “,(4,5,6), ’ , , , , fff “,(4,5,6)  
 , , , , ,  
**IDirectSound3DListener::CommitDeferredSettings** fff, , , , “,(4,5,6)  
 , , ,

---

←  
 DirectSound の  
 の

**IDirectSound3DListener::SetDistanceFactor** *fDistanceFactor*  
 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 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**

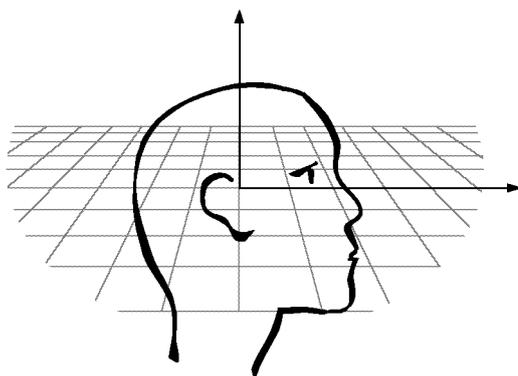
のの

**fff**□, •OE

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)

## ***f*** ***fff*** —

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

## **IDirectSoundBuffer** ***fff*** ***fff***

*IDirectSoundBuffer*

---

**Note** DirectSoundBuffer *fff*,,,, ,DirectSound *fff*, —  
,, DirectSound *fff*,%\*,,,, ,*fff*, ,*fff*,,,%\*,, Ž ,,,,,,

---

のの ののの

- Š—
- *fff*ŠŠ—
- ,Ž“
- *fff*Š—

**Š—**

の

**IDirectSoundBuffer::Play**

**IDirectSoundBuffer::Stop**

**IDirectSoundBuffer::Play**

**IDirectSoundBuffer::Stop**

**IDirectSoundBuffer::Lock**

□□□

□□□

**IDirectSoundBuffer::Unlock**

の

**IDirectSoundBuffer::GetCurrentPosition**

**IDirectSoundBuffer::SetCurrentPosition**

**ffffŠŠ—**

IDirectSoundBuffer::GetVolume      IDirectSoundBuffer::SetVolume

∞∞

IDirectSoundBuffer::GetFrequency  
IDirectSoundBuffer::SetFrequency

∞

∞      IDirectSoundBuffer::GetPan  
IDirectSoundBuffer::SetPan**□•Ž“**IDirectSoundBuffer::GetCaps      DirectSoundBuffer  
IDirectSoundBuffer::GetStatusIDirectSoundBuffer::GetFormat  
∞      IDirectSoundBuffer::GetFormat  
IDirectSoundBuffer::SetFormat

---

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

---

**fffŠ—**

IDirectSoundBuffer::Restore      DirectSoundBuffer

IDirectSoundBuffer::Restore ∞∞

∞∞

**DirectSound,Ž—**

DirectSound ∞

- 1 **DirectSoundCreateŠ** ,Œ, ,DirectSoundffffff, ,,
- 2

**IDirectSound::SetCooperativeLevel**ffff,Œ, ,<fff,Ž',, ,,,,ffff 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::Lockffff,Ž—  
, ,ffff,f f,fff,,,IDirectSoundBuffer::Unlockffff,Ž—,,
- 5 **IDirectSoundBuffer::Play**ffff,Ž—,,ffffff ffff, ,,

```

6 ffff fff,ffff, , —
  ,, DirectSoundBufferffff, IDirectSoundBuffer::Stopffff,Ž—
  ,,ffff,Ž,,
7 fffff ffff,%o*,
8 DirectSoundfffff,%o*,

```

④

- 

```

ffff ffff, , IDirectSoundBuffer::SetFormatffff,Œ, , fffff ff
ff ffff, —
ŒŽ, ', , ',', ffff fff,'fff,DSSCL_PRIORITY, ',,,,•—,,

```

- fffff ffff ffff, , IDirectSoundBuffer::Playffff,Ž—
   
,,ffff, , , , ,fffff ffff, ,,,,,, fffff ffff,,, ,,,,,,•
   
,, „ ,CPUff ,,,, ”,, fffff ffff, ,,,,,ŠŽŽŠ,' ,

④DirectSound

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

## DirectSoundfffff, □ □

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, ->, <%, ,
} APPINSTANCEDATA, *LPAPPINSTANCEDATA;
BOOL AppEnumCallbackFunction(
    LPGUID lpGuid,
    LPTSTR lpstrDescription,
    LPTSTR lpstrModule,
    LPVOID lpContext)
{
    LPAPPINSTANCEDATA lpInstance = (LPAPPINSTANCEDATA)
    lpContext;
    // lpInstance->GUID, ff, ,
    // Strcpy(->, lpInstance->,
    return TRUE; // -, ', ,
}

```

① **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**

①

**CoCreateInstance, ->, DirectSound ffffffff, □ □****CoCreateInstance**      **DirectSound**

```

1 $ ,NULL, ->, , COM, $%, ,
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□fff□``,□',,
    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□fff□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);

```



```

DSBCAPS      dwFlags      DSBCAPS_LOCHARDWARE
DSBCAPS_LOCSOFTWARE

```

```

fffff□ffff□ffff,ffff□ffff□ffff

```

**DSBUFFERDESC**

DirectSound の の の の

```

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

```

DirectSound の

**DSBPLAY\_LOOPING**

```

BOOL AppCreateWritePrimaryBuffer(
    LPDIRECTSOUND lpDirectSound, LPDIRECTSOUNDBUFFER *lpDsb,
    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(&lpDsb, 0, sizeof(DSBUFFERDESC)); // 0, □'
    dsbdesc.dwSize = sizeof(DSBUFFERDESC);
    dsbdesc.dwFlags = DSBCAPS_PRIMARYBUFFER;
    // ffffffff,ffff<Š,,,□E,,,
    dsbdesc.dwBufferBytes = 0;
    dsbdesc.lpwfxFormat = NULL; // fffff□ffff,NULL,,,,,,,,,

```

```

    // □,□,-□'fff,ž"
    hr = lpDirectSound->lpVtbl->SetCooperativeLevel(lpDirectSound,
        hwnd, DSSCL_WRITEPRIMARY);
    if(DS_OK == hr) {

```









---

# DirectSound *fffff*

Š□

## DirectSoundCreate

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

*IDirectSound* ☺

- DS\_OK ☺  
**DSERR\_ALLOCATED**  
**DSERR\_INVALIDPARAM**  
**DSERR\_NOAGGREGATION**  
**DSERR\_NODRIVER**  
**DSERR\_OUTOFMEMORY**

*lpGuid*

ffffff,Ž•,GUID,,ffff□  
,,fff f,'DirectSoundEnumerate,,•,,GUIDs,,, ,,,ffff,ffff,—◁,,,  
NULL,,

*ppDS*

„Š „%“,, ,,DirectSoundffffff,,ffff

*pUnkOuter*

—'◁,^ ,,'NULL,,,,,,

DirectSound **IDirectSound::SetCooperativeLevel**

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

## DirectSoundEnumerate

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

DirectSound

- ☺,,,\_, Ž”,,DSERR\_INVALIDPARAM,ff f f,•

*lpDSEnumCallback*

**DSEnumCallback**Š „ffff „ffff,ffff f,,,,,,DirectSoundffffff,,  
☺, ,,

*lpContext*

f f'◁ffffff,,ffff „ffffff, —◁ ffffŠ ,☺, ,,,, „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*, DirectSound*ffff*, , , , DirectSoundCreateŠ , , ,

*lpstrDescription*

DirectSound*ffff*, *ffff* < , —, , NULL, , •Ž—, *ffff*

*lpstrModule*

, *ffff*, “, , DirectSound*ffff*, *fff* f—, Ž’, , NULL, , •Ž—, *ffff*

*lpContext*

*ffff* *fff*’ < f , *ffff* , , , f *ffff*Š , “, ,

*lpstrDescription*      *lpstrModule* ののののの

**DirectSoundEnumerate****IDirectSound**

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

**IDirectSound** のの*ffff*Š, “,**Compact****Initialize***ffff***CreateSoundBuffer****DuplicateSoundBuffer****SetCooperativeLevel***ffff* ”**GetCaps***ff* f ’**GetSpeakerConfig****SetSpeakerConfig**

の COM

**IDirectSound**  
**IUnknown** の

*IUnknown*

**AddRef**  
**QueryInterface**  
**Release**

## IDirectSound::Compact

HRESULT Compact ();

- DS\_OK, DSERR\_INVALIDPARAM, DSERR\_PRIOLEVELNEEDED, DSERR\_UNINITIALIZED

の DirectSound

**IDirectSound::SetCooperativeLevel**  
DSSCL\_EXCLUSIVE) の

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

## IDirectSound::CreateSoundBuffer

HRESULT CreateSoundBuffer(LPDSBUFFERDESC lpDSBufferDesc,  
LPLPDIRECTSOUNDBUFFER lplpDirectSoundBuffer,  
IUnknown FAR \* pUnkOuter);

DirectSoundBuffer

- DS\_OK, DSERR\_ALLOCATED, DSERR\_BADFORMAT, DSERR\_INVALIDPARAM, DSERR\_NOAGGREGATION, DSERR\_OUTOFMEMORY, DSERR\_UNINITIALIZED, DSERR\_UNSUPPORTED

*lpDSBufferDesc*

DSBUFFERDESC

*lplpDirectSoundBuffer*

DirectSoundBuffer, 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

LPDSCAPS, DSCAPS

DirectSoundCreate, DSCAPS, IDirectSound

## IDirectSound::GetSpeakerConfig

HRESULT GetSpeakerConfig(LPDWORD lpdwSpeakerConfig);

DirectSound

- DS\_OK, DSERR\_INVALIDPARAM, DSERR\_UNINITIALIZED

lpdwSpeakerConfig

DS\_SPEAKER\_HEADPHONE, DS\_SPEAKER\_MONO, DS\_SPEAKER\_QUAD, DS\_SPEAKER\_STEREO, DS\_SPEAKER\_SURROUND

IDirectSound, IDirectSound::SetSpeakerConfig

## IDirectSound::Initialize

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

**CoCreateInstance** DirectSound

- E,,DS\_OK Ž”,,Ž,ff f f,•,  
**DSERR\_ALREADYINITIALIZED**  
**DSERR\_GENERIC**  
**DSERR\_INVALIDPARAM**  
**DSERR\_NODRIVER**

*lpGuid*

```
ff ffff fŽ•Ž(GUID)
,,ffff ,,Ž•Ž, ,,DirectSoundffffff,☉ ,,,,ffffff,Ž’,, 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);
```

- E,,DS\_OK Ž”,,Ž,ff f f,•,  
**DSERR\_ALLOCATED**  
**DSERR\_INVALIDPARAM**  
**DSERR\_UNINITIALIZED**  
**DSERR\_UNSUPPORTED**

*hwnd*

```
ffff fff,ffff ffff
```

*dwLevel*

```
—<,,<’fff Ž,’Ž’,,,
```

**DSSCL\_EXCLUSIVE**

```
ffff fff,“fff, ’, “—ff ff,“,, ,ffff fff,,—^
,,,,,,(DSBCAPS_GLOBALFOCUSfff ’,ffff fff,,ffff,ff f,,)
,,fff,, ffff fff, DSSCL_PRIORITYfff,,,,“☉,—
, , ,IDirectSoundBuffer::SetFormatffff,☉, ,Ž’,,, DirectSoun
```

---

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

#### DSSCL\_NORMAL

ffff fff, •, < ' ' ,,, ,ff f, ffffffff, ff f<—  
, ,, ,, ,fff fff,, fff, ' ,,,

#### DSSCL\_PRIORITY

ffff fff, —  
fff, ',,, ,, fff, ffff fff, IDirectSoundBuffer::SetFormat,, IDirectSound::Compact ffff, Ć, ,, ,,, ,,,

#### DSSCL\_WRITEPRIMARY

,,, ^, fff,, ,fff fff, fffff ffff ffff,, ,, ,fff,—  
,,, fffff ffff ffff, ; ^ ,,,

①

①

DSSCL\_NORMAL ①

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

## IDirectSound::SetSpeakerConfig

HRESULT SetSpeakerConfig(DWORD dwSpeakerConfig);

DirectSound

- Ć,, DS\_OK Ž”,, Ž, ff f f, •,

**DSERR\_INVALIDPARAM**

**DSERR\_UNINITIALIZED**

*dwSpeakerConfig*

DirectSound ffffffff, ff f ' ^%o, ' Ž, ,,,

**DSSPEAKER\_HEADPHONE**

ff f, fffff, ,,,

**DSSPEAKER\_MONO**

ff f, ffff, ,,,

**DSSPEAKER\_QUAD**

ff f, 4ffff, ,,,

**DSSPEAKER\_STEREO**

ff f, ffff, ,,, (fffff')

**DSSPEAKER\_SURROUND**

ff f, fffff ffff, ,,,

*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**

” **GetPosition**  
**SetPosition**

*ffff ffffffff* **GetConeAngles**  
*f f* **GetConeOrientation**  
**GetConeOutsideVolume**  
**SetConeAngles**  
**SetConeOrientation**  
**SetConeOutsideVolume**

“ **GetVelocity**  
**SetVelocity**

の COM **IDirectSound3DBuffer**  
*IUnknown* **IUnknown** のの

**AddRef**  
**QueryInterface**  
**Release**

### **IDirectSound3DBuffer::GetAllParameters**

`HRESULT GetAllParameters(LPDS3DBUFFER lpDs3dBuffer);`

のの 3D

- $\mathbb{E},,DS\_OK \check{Z},,,DSERR\_INVALIDPARAMff f f ,\bullet,$

*lpDs3dBuffer*

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

## **IDirectSound3DBuffer::GetConeAngles**

HRESULT GetConeAngles(  
LPDWORD lpdwInsideConeAngle, LPDWORD lpdwOutsideConeAngle);

∞∞

- $\mathbb{E},,DS\_OK \check{Z},,,DSERR\_INVALIDPARAMff f f ,\bullet,$

*lpdwInsideConeAngle lpdwOutsideConeAngle*

ffff ffffffff f f,“•Š,Š•Š,Ž,• ,ffff

## **IDirectSound3DBuffer::GetConeOrientation**

HRESULT GetConeOrientation(LPD3DVECTOR lpvOrientation);

∞∞

- $\mathbb{E},,DS\_OK \check{Z},,,DSERR\_INVALIDPARAMff f f ,\bullet,$

*lpvOrientation*

**D3DVECTOR** “,,ffff ,, “,ffff ffffffff f f,⊕ ,⊕,Ž,,, fff  
f •,ffff f f,’ •,

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

## **IDirectSound3DBuffer::GetConeOutsideVolume**

HRESULT GetConeOutsideVolume(LPLONG lpnConeOutsideVolume);

∞

- $\mathbb{E},,DS\_OK \check{Z},,,DSERR\_INVALIDPARAMff f f ,\bullet,$

*lpnConeOutsideVolume*

ffff,⊕ ,f fŠ•fff f,Ž,• ,ffff

0( ) -10,000( )∞

DirectSound

∞

**IDirectSoundBuffer::SetVolume**

## IDirectSound3DBuffer::GetMaxDistance

HRESULT GetMaxDistance(LPD3DVALUE lpflMaxDistance);

∞

- E,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAM ff f f,•,

lpflMaxDistance

E , '← ' ,Ž,• ,ffff

∅

□□□□□□□□□□□□□□□□□□

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

## IDirectSound3DBuffer::GetMinDistance

HRESULT GetMinDistance(LPD3DVALUE lpflMinDistance);

∞

- E,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAM ff f f,•,

lpflMinDistance

E , '← ' ,Ž,• ,ffff

1.0 ∞

1.0 ∅

□□□□□□□□□□□□□□□□□□

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

## IDirectSound3DBuffer::GetMode

HRESULT GetMode(LPDWORD lpdwMode);

3D ∞

- E,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAM ff f f,•,

lpdwMode

E , f f,Ž,• ,ffff Ž,’,•,

DS3DMODE\_DISABLE

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

DS3DMODE\_HEADRELATIVE

ffffff f(^ “ •E)

,fff ,fff f,“ŠE,,, ,f f,, ffff, ‘fff f,fff fff f,• ,•,•,•,

Ž““ , ,•,•,• “fff f,•%0,•,•,•,•,•,

DS3DMODE\_NORMAL

• , — ,,,ffffff f,,,,,

## IDirectSound3DBuffer::GetPosition

HRESULT GetPosition(LPD3DVECTOR lpvPosition);

∞∞

- E,,DS\_OK Ž”,,DSERR\_INVALIDPARAMff f f,\*

lpvPosition

D3DVECTOR “,,ffff , “,ffff ffff’,ffff fffff,Ž,,,

## IDirectSound3DBuffer::GetVelocity

HRESULT GetVelocity(LPD3DVECTOR lpvVelocity);

∞∞

- E,,DS\_OK Ž”,,DSERR\_INVALIDPARAMff f f,\*

lpvVelocity

D3DVECTOR “,,ffff , “,ffff ffff,E “,Ž,,,

∅

IDirectSound3DBuffer::SetPosition,  
IDirectSound3DBuffer::SetVelocity

## IDirectSound3DBuffer::SetAllParameters

HRESULT SetAllParameters(  
LPDS3DBUFFER lpDs3dBuffer, DWORD dwApply);

∞∞∞

DS3DBUFFER

∅ 3D

- E,,DS\_OK Ž”,,DSERR\_INVALIDPARAMff f f,\*

lpDs3dBuffer

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

dwApply

,fffff,Ž,’ Ž,’Ž,,,

DS3D\_DEFERRED

ffff fff,IDirectSound3DListener::Com  
mitDeferredSettingsffff,E, ,,’ , ,,,

• , ’,\* ,,, “, EŽ,Ž ,,,,%”

DS3D\_IMMEDIATE

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

## IDirectSound3DBuffer::SetConeAngles

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

○○○

- E,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAMff f f,\*,

*dwInsideConeAngle*      *dwOutsideConeAngle*  
ffff ffffffff f f,“Š,Š•Š

*dwApply*  
, ,,,fffff,Ž,’ Ž,’Ž,,,

DS3D\_DEFERRED      ffff fff, IDirectSound3DListener::Com  
mitDeferredSettingsffff,E, ,,, ’, ,,,,  
• , ’,• ,,, “, EŽ,Ž ,,,,%”

DS3D\_IMMEDIATE      ’,Ž, ,, ffff,,,,,3Dffff ffff,‘,,3D •  
, EŽ, ,

○ 0 (      )      360 (      )      360

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

## IDirectSound3DBuffer::SetConeOrientation

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

○○○

- E,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAMff f f,\*,

*x, y, z*  
D3DVALUE,fff,Ž,’ ,,, <,ffff f f,•Effff, •,•,

*dwApply*  
, ,,,fffff,Ž,’ ~%o,’Ž,,,

DS3D\_DEFERRED      ffff fff, IDirectSound3DListener::Comm  
itDeferredSettingsffff,E, ,,, ’, ,,,, •  
, ’,• ,,, “, EŽ,Ž ,,,,%”

DS3D\_IMMEDIATE      ’,Ž, ,, ffff,,,,,3Dffff ffff,‘,,3D •  
, EŽ, ,

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

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





**DS3D\_DEFERRED**

*ffff fff, IDirectSound3DListener::Com  
mitDeferredSettingsffff, Ć, ,, ', ,,,  
• , ',• ,,, " ĆŽ,Ž ,,,,‰"*

**DS3D\_IMMEDIATE**

*' ,Ž, ,, ffff,,,,,3Dffff ffff,' ,3D •  
, ĆŽ, ,*

## **IDirectSound3DBuffer::SetPosition**

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

∞∞

**IDirectSound3DListener::SetDistanceFactor**

- Ć,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAMff f f,•,

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

*dwApply  
' , ,,,fffff,Ž,' ^‰,Ž,,*

**DS3D\_DEFERRED**

*ffff fff, IDirectSound3DListener::Com  
mitDeferredSettingsffff, Ć, ,, ', ,,,  
• , ',• ,,, " ĆŽ,Ž ,,,,‰"*

**DS3D\_IMMEDIATE**

*' ,Ž, ,, ffff,,,,,3Dffff ffff,' ,3D •  
, ĆŽ, ,*

## **IDirectSound3DBuffer::SetVelocity**

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

∞∞

- Ć,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAMff f f,•,

*x, y, z  
D3DVALUE,fff,Ž,' ,,, ,,“ffff, •,•,*

*dwApply  
' , ,,,fffff,Ž,' ^‰,Ž,,*

**DS3D\_DEFERRED**

*ffff fff, IDirectSound3DListener::Com  
mitDeferredSettingsffff, Ć, ,, ', ,,,  
• , ',• ,,, " ĆŽ,Ž ,,,,‰"*

**DS3D\_IMMEDIATE**

*' ,Ž, ,, ffff,,,,,3Dffff ffff,' ,3D •  
, ĆŽ, ,*

∞

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

## IDirectSound3DListener

の 3D の

*IDirectSound3DListener* の の の の の *IDirectSound3DListener*

*IDirectSound3DListener* の の

<i>fffffff f'</i>	<b>GetAllParameters</b> <b>SetAllParameters</b>
<i>,', '</i>	<b>CommitDeferredSettings</b>
<i>←—'</i>	<b>GetDistanceFactor</b> <b>SetDistanceFactor</b>
<i>ffff —'</i>	<b>GetDopplerFactor</b> <b>SetDopplerFactor</b>
<i>•⊕</i>	<b>GetOrientation</b> <b>SetOrientation</b>
<i>~</i>	<b>GetPosition</b> <b>SetPosition</b>
<i>f fff—'</i>	<b>GetRolloffFactor</b> <b>SetRolloffFactor</b>
<i>“</i>	<b>GetVelocity</b> <b>SetVelocity</b>

の COM  
*IUnknown*

*IDirectSound3DBuffer*  
*IUnknown* の の

**AddRef**  
**QueryInterface**  
**Release**

## *IDirectSound3DListener* **::CommitDeferredSettings**

`HRESULT CommitDeferredSettings();`

ののの

- `DS_OK` `DSERR_INVALIDPARAM` `ff f f,•`,

のの

□□□□□□□□□□□□□□

## **IDirectSound3DListener::GetAllParameters**

`HRESULT GetAllParameters(LPD3DLISTENER lpListener);`

3D 3D のの

- `DS_OK` `DSERR_INVALIDPARAM` `ff f f,•`,

*lpListener*

`DS3DLISTENER` “,ffff ,, “,3Df ff,,3Dfff ,`DS` , ‘,Ž,

**IDirectSound3DListener::SetAllParameters**

## **IDirectSound3DListener::GetDistanceFactor**

`HRESULT GetDistanceFactor(LPD3DVALUE lpflDistanceFactor);`

の

- `DS_OK` `DSERR_INVALIDPARAM` `ff f f,•`,

*lpflDistanceFactor*

`D3DVALUE`fff,• ,,ffff ,,`DS` ,←—‘,’Ž,,

**IDirectSound3DListener::SetDistanceFactor**

## **IDirectSound3DListener::GetDopplerFactor**

`HRESULT GetDopplerFactor(LPD3DVALUE lpflDopplerFactor);`

の

- `DS_OK` `DSERR_INVALIDPARAM` `ff f f,•`,

*lpflDopplerFactor*

`D3DVALUE`fff,• ,,ffff ,,`DS` ffff —‘,’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

- $\mathbb{E},,,\text{DS\_OK}$   $\checkmark$ ,,, $\text{DSERR\_INVALIDPARAM}$ 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**

- $\mathbb{E},,,\text{DS\_OK}$   $\checkmark$ ,,, $\text{DSERR\_INVALIDPARAM}$ ff f f,•,

*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,•,

*lpflRolloffFactor*

**D3DVALUE**fff,• ,ffff  $\mathbb{E}$  ,f fff—‘,’,Ž,,,,,

**DS3D\_DEFAULTROLLOFFFACTOR** (1.0)

**IDirectSound3DListener::SetRolloffFactor**



DS3D\_IMMEDIATE **mitDeferredSettings**ffff,CE, ,,, ', ,,,,  
 • , ',• ,,, "“, CEŽ,Ž ,,,,,%o",,,  
 ',Ž, ,, ffff,,,,,3Dffff ffff,' ,3D •  
 , CEŽ, ,

**IDirectSound3DListener::GetDistanceFactor**

**IDirectSound3DListener::SetDopplerFactor**

```
HRESULT SetDopplerFactor(
    D3DVALUE f1DopplerFactor, DWORD dwApply);
```

のの

- CE,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAMff f f ,•,

*f1DopplerFactor*  
 ,,ffff —‘,’

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

DS3D\_DEFERRED **ffff fff,IDirectSound3DListener::Com  
mitDeferredSettings**ffff,CE, ,,, ', ,,,,  
 • , ',• ,,, "“, CEŽ,Ž ,,,,,%o",,,  
 ',Ž, ,, ffff,,,,,3Dffff ffff,' ,3D •  
 , CEŽ, ,

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

- CE,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAMff f f ,•,

*xFront, yFront, zFront*  
**D3DVALUE**fff,' ‘ffff, •,Ž,,,

*xTop, yTop, zTop*  
**D3DVALUE**fff,' •ffff, •,Ž,,,

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

DS3D_DEFERRED	ffff fff, IDirectSound3DListener::Com mitDeferredSettingsffff, E, ,, ', ,,, • , ',• ,,, "“, ĆŽ,Ž ,,,,,%o",,,
DS3D_IMMEDIATE	', Ž, ,, ffff,,,,, 3Dffff ffff, ',, 3D • , ĆŽ, ,
(0,0,1.0)	(0,1.0,0)

### IDirectSound3DListener::GetOrientation

## IDirectSound3DListener::SetPosition

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

### ○の○のIDirectSound3DListener::SetDistanceFactor

- Ć,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAMff f f,•,

*x, y, z*  
D3DVALUEfff, ' fff , < ^'ffff, •, Ž,,,

*dwApply*  
' , ,,,fffff,Ž, ' ~%o, ' Ž,,,

DS3D_DEFERRED	ffff fff, IDirectSound3DListener::Com mitDeferredSettingsffff, E, ,, ', ,,, • , ',• ,,, "“, ĆŽ,Ž ,,,,,%o",,,
DS3D_IMMEDIATE	', Ž, ,, ffff,,,,, 3Dffff ffff, ',, 3D • , ĆŽ, ,

### IDirectSound3DListener::GetPosition

## IDirectSound3DListener::SetRolloffFactor

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

- Ć,,,DS\_OK Ž”,,,DSERR\_INVALIDPARAMff f f,•,

*flRolloffFactor*  
<, f fff—‘

*dwApply*  
' , ,,,fffff,Ž, ' ~%o, ' Ž,,,

DS3D_DEFERRED	ffff fff, IDirectSound3DListener::Commi tDeferredSettingsffff, E, ,, ', ,,, • , ',• ,,, "“, ĆŽ,Ž ,%o",,,
DS3D_IMMEDIATE	', Ž, ,, ffff,,,,, 3Dffff ffff, ',, 3D •



**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

- 0, 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

**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);`



**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);

- 0, DS\_OK Ž” ,Ž, f 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);

の

- 0, DS\_OK Ž” ,Ž, **DSERR\_INVALIDPARAM** f f f ,•

*lpdwStatus*  
ffff ffff, ‘Ž,• ,ffff ‘Ž,’ ,’  
**DSBSTATUS\_BUFFERLOST**  
ffff,Ž,,,, ,fff, ;ffff,•—,,



---

## IDirectSoundBuffer::Lock

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

の

- DS\_OK, DSERR\_BUFFERLOST, DSERR\_INVALIDCALL, DSERR\_INVALIDPARAM, DSERR\_PRIOLEVELNEEDED

**DSERR\_BUFFERLOST**  
**DSERR\_INVALIDCALL**  
**DSERR\_INVALIDPARAM**  
**DSERR\_PRIOLEVELNEEDED**

*dwWriteCursor*

ffff,ŠŽ^,,fff,ŠŽ,,^,,, fff^,fffff  
dwFlagsfff f,DSBLOCK\_FROMWRITECURSORfff,Ž',,,,, , ,fff  
f,-Ž,,

*dwWriteBytes*

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

*lplpvAudioPtr1*

fff,,ffff, ,ffff,ffff

*lpdwAudioBytes1*

lplpvAudioPtr1fff f,Ž,,fff ,Ž,• ,ffff ,,',dwWriteBytesfff f,, ,,  
lplpvAudioPtr2,ffffff f,Ž,ffff,Ž,

*lplpvAudioPtr2*

fff,,ffff ffff,2”—,—^,ffff ,,fff f,'NULL,  
lplpvAudioPtr1fff f,ffff ffff,fff,,—^“;Ž,

*lpdwAudioBytes2*

lplpvAudioPtr2 fff f,ffff,,fff ,Ž,• ,ffff  
lplpvAudioPtr2,NULL, ,,',0,,

*dwFlags*

fff,ffff, ,,fff ^%o,'Ž,,

**DSBLOCK\_FROMWRITECURSOR**

☺ , ,,,f ff,fff, ,  
,,,,,IDirectSoundBuffer::GetCurrentPosition,☺, ,,•—  
,, ,fff,Ž',,,,dwWriteCursorfff f,-Ž,, ,fff,”^Ž',,,

のののののの *lplpvAudioBytes2* NULL のの

*lplpvAudioPtr2*

*lpdwAudioBytes2*

NULL

DirectSound の

**IDirectSoundBuffer::Lock**

DirectSound

**IDirectSoundBuffer::Unlock**

**Warning**

DirectSoundBuffer::Unlock

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

**IDirectSoundBuffer::Play**

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

○

- **DS\_OK**
- **DSERR\_BUFFERLOST**
- **DSERR\_INVALIDCALL**
- **DSERR\_INVALIDPARAM**
- **DSERR\_PRIOLEVELNEEDED**

*dwReserved1*

Reserved1

*dwReserved2*

Reserved2

*dwFlags*

DSBPLAY\_LOOPING

**DSBPLAY\_LOOPING**

Looping flag

○

□□□□□□□□□□

DSBPLAY\_LOOPING ○

∞





---

`,0, Ž” ,CE ,ffffCEŽ,ffff,, ,CEŽ, IDirectSound::CreateSoundBufferffff,Ž’,,,,,`

のののの

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

## IDirectSoundBuffer::SetPan

`HRESULT SetPan(LONG lPan);`

- `CE,,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

- `CE,,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,CE ,•,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::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

- 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, lpvAudioPtr2 fff f,Ž“ ,,,• ,fff
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**

,,,ffffffff 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

ffffffff 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

①

#### 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,Direct  
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,ffff ffff,,ffff,,,,—%o”,,,,,,• ,,•—,,

#### DSBCAPS\_LOCSOFTWARE

DSBCAPS\_STATIC ,Ž’,,f ffff ff f,—

```

%”,,,,, ffff,“ fffffff 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,Ⓔ, ,,Ž’,,,,,,
,,fff,fff,,, ,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 fŠ,ffff fff ,,,,%o%o
(ActiveMovie™),,—, ,,, f f,,,’DirectSound
ffff fff, ,’, ,* ,ffff ff ff—,ffff ffff,,, %o,,,

```

**dwBufferBytes**

```

,,ffff,fff^,fff

```

**dwUnlockTransferRate**

```

IDirectSoundBuffer::Unlock,Ⓔ, ,,,,,ffff fff,f f,“ ,,,f f, ,*—
,kfff,Ž’,, fffff fff,ffff fff, IDirectSoundBuffer::Unlock,Ž ,,,,,
—,ŽŠ,Ⓔ’,,,,, ,’,Ž,,,,, fffffff,’,,,,,ffffff ffff,,,f f, —,*
—,,” , ,
f ffff ffff,,,f f, ffff,ffff f f,f f,,,%o” ,,,,, ,’,,%o” ,,, ,
,,“““‘Ⓔ’, ,,, ,,,

```

**dwPlayCpuOverhead**

```

,,ffff ffff,ffffff,,,•—,f f fff,Ž, —ffff,f fff f,Ž’,,
f ffff ffff,,,, ,fff,ffffff,ffff ffff,,, ,,,,, ,,,
ffffff ffff,,, ,fff,ffff ff fff,ffff fffff,“^’,,

```

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

①

<b>DSBCAPS</b>	<b>dwFlags</b>	<b>DSBUFFERDESC</b>
		① <b>DSBCAPS</b>
<b>DSBCAPS_LOCHARDWARE</b>		<b>DSBCAPS_LOCSOFTWARE</b>
	①	<b>DSBUFFERDESC</b>
∞∞∞		

**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 ,%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,Direct  
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,—
%o",,,,,, 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,Ā, ,,'Ž',,,,,,
,,fff,fff,,, ,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 ,,'ffffff ffff, ,,,,,,,,,,,,,,
```

**dwReserved**

```
,,fff,—,,,, Ž—,,,,
```

**lpwfxFormat**

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

**dwFlags** DSBCAPS\_LOCHARDWARE

DSBCAPS\_LOCSOFTWARE

DSBCAPS\_LOCHARDWARE

Ⓞ

DSBCAPS\_LOCSOFTWARE

Ⓞ **DSBCAPS** **dwFlags**  
DirectSoundBuffer

**dwBufferBytes**

DirectSound ⓄⓄ

ⓄⓄ

**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

⑦

### dwSize

„“,fff”^,fff

### dwFlags

ffff,”—,Ž’,, ^%o,,,,,,• ,Ž’,,,

#### DSCAPS\_CERTIFIED

„ffff,Microsoft,,,ŽĀ,,• ,,,,

#### DSCAPS\_CONTINUOUSRATE

ffff,dwMinSecondarySampleRate ,dwMaxSecondarySampleRate  
 fff’,Š,,,,,ffffffffff 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,,ffffffffff,ffffff ffff ffff,ff f,,

#### DSCAPS\_PRIMARY8BIT

ffff,,ffffffffff,ffffff ffff,ff f,,

#### DSCAPS\_PRIMARYMONO

ffff,%offffff ffff,ff f,,

#### DSCAPS\_PRIMARYSTEREO

ffff,ffffffffff ffff,ff f,,

#### DSCAPS\_SECONDARY16BIT

ffff,16ffffffffff,f ffff fffff fffff ffff,ff f,,

#### DSCAPS\_SECONDARY8BIT

```

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

```



”  
**DSERR\_UNSUPPORTED**  
E, „Š ,,,,ff f,,,,,