

Part C

Microsoft® DirectX™ 3
ffffffŠ”fff

„fffff, < ,,,, •, — ,,,• ,,,,,, ,, •,%Ž- -
,,f f,“ < ,,,,Œ,%<,,,,, ,,fffff,‘•,,^•, Microsoft Corporation, —,,%
,,, “<“ <Š“ ,,,,•Ž •-,,• “ ,,,,,,
Microsoft , ,,fffff, < ,,,, fff fff,“ <,,, ’,“ < • ’ Œ ,,’“ —
Œ,•Ž,,, ,,fffff, 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, •,,

Direct3D' f f ffffff

fff.....
f ffffŠ
IDirect3D.....
IDirect3DDevice.....
IDirect3DExecuteBuffer.....
IDirect3DLight.....
IDirect3DMaterial.....
IDirect3DTexture.....
IDirect3DViewport.....
“.....
—(E.....
,,‘E.....
•’.....

fff

D3DDivide

D3DDivide(a, b) (float)((double)(a) / (double)(b))

- Š,Ž, ,•,

a^b

Ž,•,•Ž

D3DMultiply

D3DMultiply

D3DMultiply(a, b) ((a) * (b))

- 2', ,•,

a^b

Š, ,,,,'

D3DDivide

D3DRGB

D3DRGB(r, g, b) \
(0xff000000L | (((long)((r) * 255) << 16) | \
(((long)((g) * 255) << 8) | (long)((b) * 255))

RGB の

- RGB •',D3DCOLOR',•Š,,

r, g, b

, (red) —(green),,, (blue),—' ,,,,', 0,,1,,",•“ “€,,

D3DRGBA

D3DRGBA

D3DRGBA(r, g, b, a) \
(((long)((a) * 255) << 24) | ((long)((r) * 255) << 16) | \
(((long)((g) * 255) << 8) | (long)((b) * 255))

RGBA の

- RGBA •',D3DCOLOR',•Š,,

r, g, b, a
 , (red) —(green) (blue),,,ffff(alpha),—
D3DRGB

D3DSTATE_OVERRIDE

D3DSTATE_OVERRIDE(*type*) ((DWORD) (*type*) + D3DSTATE_OVERRIDE_BIAS)

ののの

- ‘,’,’,’,

type

*f f fff, ‘ ,fff f,D3DTRANSFORMSTATETYPE D3DLIGHTS
 TATETYPE D3DRENDERSTATETYPE*€,fff,,,,,,
 STATE_DATA (DirectX SDK のの Misc/D3dmacs.h
) D3DSTATE_OVERRIDE D3DRENDERSTATE_SHADEMODE
 ののの

// -`f□f,fff

STATE_DATA(D3DSTATE_OVERRIDE(D3DRENDERSTATE_SHADEMODE), TRUE,
 lpBuffer);

// f□ffff□,,,,,□-`f□f,ž□,fff%□,,

STATE_DATA(D3DSTATE_OVERRIDE(D3DRENDERSTATE_SHADEMODE), FALSE,
 lpBuffer);

ののの

D3DVAL

D3DVAL(*val*) ((float)*val*)

D3DVALUE の

- Š€,’,*,

val

- Š,,,’

D3DVALP

D3DVALP

D3DVALP(*val, prec*) ((float)*val*)

- `•ŠE,’,•,`

val
`•Š,,,’`

prec
`-E`

D3DVAL

16

D3DVAL

RGB_GETBLUE

`RGB_GETBLUE(rgb) ((rgb) & 0xff)`

D3DCOLOR

- `•,•,`

rgb
`•,ž, , ,D3DCOLOR’`

RGB_GETGREEN

`RGB_GETGREEN(rgb) (((rgb) >> 8) & 0xff)`

D3DCOLOR

- `— •,•,`

rgb
`— •,ž, , ,D3DCOLOR’`

RGB_GETRED

`RGB_GETRED(rgb) (((rgb) >> 16) & 0xff)`

D3DCOLOR

- `•,•,`

rgb
`•,ž, , ,D3DCOLOR’`

RGB_MAKE

`RGB_MAKE(r, g, b) ((D3DCOLOR) (((r) << 16) | ((g) << 8) | (b)))`

RGB

- `•,’,•,`

r, g, b
 ,, , — ,—‘ ’”^,0,,255

RGB_TORGBA

RGB_TORGBA(*rgb*) ((D3DCOLOR) ((*rgb*) | 0xff000000))

RGB RGBA

- RGBA ,•,

rgb
 RGBA ,•Š,,RGB
RGBA_TORGB

RGBA_GETALPHA

RGBA_GETALPHA(*rgb*) ((*rgb*) >> 24)

RGBA D3DCOLOR

- *ffff* •’,•,

rgb
ffff •,Ž, , ,D3DCOLOR’

RGBA_GETBLUE

RGBA_GETBLUE(*rgb*) ((*rgb*) & 0xff)

RGBA D3DCOLOR

- •’,•,

rgb
 •,Ž, , ,D3DCOLOR’

RGBA_GETGREEN

RGBA_GETGREEN(*rgb*) (((*rgb*) >> 8) & 0xff)

RGBA D3DCOLOR

- — •’,•,

rgb
 — •,Ž, , ,D3DCOLOR’

RGBA_GETRED

RGBA_GETRED(*rgb*) (((*rgb*) >> 16) & 0xff)

RGBA D3DCOLOR

- ,;

rgb
,, , , D3DCOLOR'

RGBA_MAKE

```
RGBA_MAKE(r, g, b, a) \
((D3DCOLOR) (((a) << 24) | ((r) << 16) | ((g) << 8) | (b)))
```

RGBA D3DCOLOR

- ,;

r, g, b, a
,, RGBA , — *ffff* •'

RGBA_SETALPHA

```
RGBA_SETALPHA(rgba, x) (((x) << 24) | ((rgba) & 0x00ffffff))
```

RGBA D3DCOLOR

- *ffff* •', ',, RGBA ,;

rgba
ffff •', ',, RGBA

x
,, *ffff* •'

RGBA_TORGB

```
RGBA_TORGB(rgba) ((D3DCOLOR) ((rgba) & 0xffffffff))
```

RGBA D3DCOLOR

RGB D3DCOLOR

- RGB ,;

rgba
RGB , , , , RGBA
RGB_TORGBA

f *ffff* *Š*

D3DENUMDEVICESCALLBACK

```
typedef HRESULT (FAR PASCAL * LPD3DENUMDEVICESCALLBACK)
(LPGUID lpGuid, LPSTR lpDeviceDescription,
```

```
LPSTR lpDeviceName, LPD3DDEVICEDESC lpD3DHWDeviceDesc,
LPD3DDEVICEDESC lpD3DHELDeviceDesc, LPVOID lpUserArg);
```

Direct3D の

- `ffff fff, ^%o,',,,,,*,`
D3DENUMRET_CANCEL
`—<,'Ž,,`
- **D3DENUMRET_OK**
`—<,'E',,`

lpGuid

`ff ff ff fŽ•Ž(GUID),,ffff`

lpDeviceDescription

`ffff,'Ž,,ffff`

lpDeviceName

`ffff-,ffff`

lpD3DHWDeviceDesc

`Direct3Dffff f ffff •,D3DDEVICEDESC “,,ffff`

lpD3DHELDeviceDesc

`Direct3Dffff ffff fff •,D3DDEVICEDESC “,,ffff`

lpUserArg

`„f ffffŠ ;“,,ffff fff’<f f,,ffff`

□□

の

D3DENUMTEXTUREFORMATSCALLBACK

```
typedef HRESULT (WINAPI* LPD3DENUMTEXTUREFORMATSCALLBACK)
(LPDDSURFACEDESC lpDdsd, LPVOID lpUserArg);
```

の

lpDdsd

`fffff •,Š,DirectDrawSurfaceffffff,,ffff`

lpUserArg

`„f ffffŠ ;“,,ffff fff’<f f,,ffff`

□□

の

D3DVALIDATECALLBACK

```
typedef HRESULT (WINAPI* LPD3DVALIDATECALLBACK)
(LPVOID lpUserArg, DWORD dwOffset);
```

IDirect3DExecuteBuffer::Validate

□□

の

□□

lpUserArg
 ,,f ffffŠ ;“,,,ffff fff’<f f,,ffff
dwOffset
 ffff,ff ,Œ ,,Ž ffff,ffff

の

IDirect3D

IDirect3D の Direct3D の IDirect3D の IDirect3D

IDirect3D の

CreateLight
CreateMaterial
CreateViewport

—<
Š%

EnumDevices
FindDevice
Initialize

IDirect3D
IUnknown

の COM の
IUnknown の

AddRef
QueryInterface
Release

IDirect3D::CreateLight

HRESULT CreateLight(LPDIRECT3DLIGHT* lpDirect3DLight,
 IUnknown* pUnkOuter);

Direct3DLight の IDirect3DViewport::AddLight

- Œ,,,D3D_OK Ž”,,, Ž’,,,,,,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpDirect3DLight
 Œ,,, IDirect3DLight fff ffff,,ffff, ’,,ffff

pUnkOuter
 ,,fff f,COM %o<”, —
 ĆŠ ,Ž,,,,,, IDirect3D::CreateLightffff,,fff f,NULL^Š,,,ff ,•,

IDirect3D::CreateMaterial

```
HRESULT CreateMaterial(LPDIRECT3DMATERIAL* lpDirect3DMaterial,
    IUnknown* pUnkOuter);
```

Direct3DMaterial

- Ć,,, D3D_OK,•, •,^—, Direct3D ’ f f,•’ ,Ž ,,,,

lpDirect3DMaterial

Ć, ,, Ć,,, IDirect3DMaterial fff ffff,,ffff,ffff

pUnkOuter

„fff f, COM“, —
 Š’,,,,,,, IDirect3D::CreateMaterialffff, ,,fff f,NULL^Š,,,ff ,•
 ,

IDirect3D::CreateViewport

```
HRESULT CreateViewport(LPDIRECT3DVIEWPORT* lpD3DViewport,
    IUnknown* pUnkOuter);
```

Direct3DViewport ①IDirect3DDevice::AddViewport

Direct3D

- Ć,,,D3D_OK Ž”,,, Ž’,,,,,,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpD3DViewport

Ć, ,, Ć,,, IDirect3DViewport fff ffff,,ffff,ffff

pUnkOuter

„fff f, COM“, —
 Š’,,,,,,, IDirect3D::CreateViewportffff, ,,fff f,NULL^Š,,,ff ,•
 •,

IDirect3D::EnumDevices

```
HRESULT EnumDevices(LPD3DENUMDEVICESCALLBACK lpEnumDevicesCallback,
    LPVOID lpUserArg);
```

Direct3D

- Ć,,,D3D_OK Ž”,,, Ž’,,,,,,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpEnumDevicesCallback

Ⓔ ,,,,Ⓔ, ,,,—< —,D3DENUMDEVICESCALLBACK
f ffffŠ ,ffff

lpUserArg

f ffffŠ ,“,ffff fff’<f f,,ffff

IDirect3D::FindDevice

HRESULT FindDevice(LP3DFINDDEVICESEARCH lpD3DFDS,
LP3DFINDDEVICERESULT lpD3DFDR);

のの

- Ⓔ,,D3D_OK,•,, •’,,,,, Direct3D ’ f f,•’ ,Ž ,,,,

lpD3DFDS

Ž“,ffff,Ž,D3DFINDDEVICESEARCH “,ffff

lpD3DFDR

Ⓔ Ž,ffff,Ž,D3DFINDDEVICERESULT “,ffff

IDirect3D::Initialize

HRESULT Initialize(REFIID lpREFIID);

の COM ののの

- Direct3Dffffff, ,,,,, Š%o
,,,, DDERR_ALREADYINITIALIZED,•,

lpREFIID

fff ff ff fŽ•Ž(UUID),,ffff,•,

IDirect3DDevice

IDirect3DDevice のDirect3D ののIDirect3DDevice のIDirect3DDevice

IDirect3DDevice のの

Ž

CreateExecuteBuffer

Execute

•

EnumTextureFormats

GetCaps

GetDirect3D

GetPickRecords

GetStats

| | |
|------------------------|----------------------------------------|
| ”_ | CreateMatrix |
| | DeleteMatrix |
| | GetMatrix |
| | SetMatrix |
| ”‘ | Initialize |
| | Pick |
| | SwapTextureHandles |
| <i>f f</i> | BeginScene |
| | EndScene |
| <i>ff f f</i> | AddViewport |
| | DeleteViewport |
| | NextViewport |
| IDirect3DDevice | Ⓞ COM |
| <i>IUnknown</i> | IUnknown Ⓞ |
| AddRef | |
| QueryInterface | |
| Release | |
| Direct3DDevice | 3D |
| | DirectDrawSurface |
| | IDirect3DDevice::QueryInterface |

IDirect3DDevice::AddViewport

```
HRESULT AddViewport(LPDIRECT3DVIEWPORT lpDirect3DViewport);
```

Ⓞ

- $\mathbb{E},,D3D_OK$ Ž”,, Ž’,,,,,,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpDirect3DViewport

Direct3DDeviceffffff,”,,IDirect3DViewportfff ffff,ffff

IDirect3DDevice::BeginScene

```
HRESULT BeginScene();
```

- $\mathbb{E},,D3D_OK,•,$

lpArg
f ffffŠ ,“,,ffff fff’<f f,,ffff

IDirect3DDevice::Execute

HRESULT Execute(LPDIRECT3DEXECUTEBUFFER lpDirect3DExecuteBuffer,
LPDIRECT3DVIEWPORT lpDirect3DViewport, DWORD dwFlags);

- Ć,,D3D_OK Ž”,, Ž,’,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpDirect3DExecuteBuffer
Ž ,,Ž ffff,,ffff

lpDirect3DViewport
•Ž,,Ž ffff,•Šffffff,Ā ,Direct3DViewportffffff,,ffff

dwFlags
ffff,ffffff,ffffff,,Ž’,,fff ,,fff f, Ž,’,,,,,,

D3DEXECUTE_CLIPPED

ff f f,Š,,ffffff,ffffff,,

D3DEXECUTE_UNCLIPPED

ff f f,’,,,,ffffff,Š,(ffffff,,)

D3DEXECUTEDATA D3DINSTRUCTION

IDirect3DExecuteBuffer::Validate

IDirect3DDevice::GetCaps

HRESULT GetCaps(LP3DDDEVICEDESC lp3DHWDevDesc,
LP3DDDEVICEDESC lp3DHELDevDesc);

Direct3DDevice 0

- Ć,,D3D_OK Ž”,, Ž,’,,,,,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpD3DHWDevDesc
ffff,f ffff •,Š,D3DDEVICEDESC “,,ffff

lpD3DHELDevDesc
—,,,,,ffffff,ffff fff •,Š,D3DDEVICEDESC “,,ffff

000**IDirectDraw2::GetCaps**

DDERR_INVALIDPARAMS

lpD3DStats
•,Ž,D3DSTATS “,,ffff

IDirect3DDevice::Initialize

HRESULT Initialize(LPDIRECT3D lpd3d, LPGUID lpGUID,
LPD3DDEVICEDESC lpd3ddvdesc);

- Ć,,D3D_OK,•, ,,‘,,,ff ,, ,’,,,,, Direct3D ’ f f,•’ ,Ž ,,,,

lpd3d
Š%o,,Direct3D ffff,,ffff

lpGUID
fff ffffŽ•Ž,,ff ff ff fŽ•Ž(GUID),,ffff

lpd3ddvdesc
Š%o,,Direct3DDeviceffffff,Ž,D3DDEVICEDESC “,,ffff

IDirect3DDevice::NextViewport

HRESULT NextViewport(LPDIRECT3DVIEWPORT lpDirect3DViewport,
LPDIRECT3DVIEWPORT* lpDirect3DViewport, DWORD dwFlags);

- Ć,,D3D_OK Ž”,,, Ž,’,,,,,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpDirect3DViewport
Direct3DDeviceffffff,Š~,,ff f f,fff,’ff f f,,ffff

lpDirect3DViewport
Direct3DDeviceffffff,Š~,,ff f f,fff,’Ž,ff f f,,ffff

dwFlags
ff f f,fff,Ž“,,ff f f,Ž’,,fff ffff,D3DNEXT_NEXT
D3DNEXT_HEAD fff, , -Ž“
D3DNEXT_NEXT fff,Ž, -Ž“
D3DNEXT_TAIL fff, Ć, -Ž“

IDirect3DDevice::Pick

HRESULT Pick(LPDIRECT3DEXECUTEBUFFER lpDirect3DExecuteBuffer,
LPDIRECT3DVIEWPORT lpDirect3DViewport, DWORD dwFlags,
LPD3DRECT lpRect);

lpRect の *z*

Direct3DExecuteBuffer の

- `D3D_OK`、`D3DERR_EXECUTE_LOCKED`、`DDERR_INVALIDOBJECT`、`DDERR_INVALIDPARAMS`

lpDirect3DExecuteBuffer

`LPDIRECT3DEXECUTEBUFFER`

lpDirect3DViewport

`LPDIRECT3DVIEWPORT`

dwFlags

`D3DVIEWPORT_FLAG_*`

lpRect

`RECT`

lpRect の *x1* *x2*

y1 *y2*

Direct3DExecuteBuffer の Direct3DDevice

IDirect3DDevice::GetPickRecords

IDirect3DDevice::SetMatrix

`HRESULT SetMatrix(D3DMATRIXHANDLE d3dMatHandle, LPD3DMATRIX lpD3DMatrix);`

この **IDirect3DDevice::CreateMatrix**

- `D3D_OK`、`DDERR_INVALIDPARAMS`

d3dMatHandle

`D3DMATRIXHANDLE`

lpD3DMatrix

`LPD3DMATRIX`

この **IDirect3DDevice::SetMatrix**

□□□□□

の

IDirect3DDevice::CreateMatrix ***IDirect3DDevice::GetMatrix***
IDirect3DDevice::DeleteMatrix

IDirect3DDevice::SwapTextureHandles

`HRESULT SwapTextureHandles(LPDIRECT3DTEXTURE lpD3DTex1, LPDIRECT3DTEXTURE lpD3DTex2);`

*IDirect3DExecuteBuffer::SetExecuteData***IDirect3DExecuteBuffer::Initialize**

```
HRESULT Initialize(LPDIRECT3DDEVICE lpDirect3DDevice,
                  LPD3DEXECUTEBUFFERDESC lpDesc);
```

COM の

- Direct3DExecuteBufferffff, Ž, Š%
,,,,,DDERR_ALREADYINITIALIZED,•

lpDirect3DDevice

Direct3D fffffff,Ž,ffff,,ffff

lpDesc

```
,,,Direct3DExecuteBufferffffff,< ,,D3DEXECUTEBUFFERDESC  
“,,ffff ffff,fff,‘,,,,, Ć, ,,Ž”,,
```

IDirect3DExecuteBuffer::Lock

```
HRESULT Lock(LPD3DEXECUTEBUFFERDESC lpDesc);
```

のの

- Ć,,D3D_OK Ž”,,, Ž’,,,,,,•,
D3DERR_EXECUTE_LOCKED
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_WASSTILLDRAWING

lpDesc

```
D3DEXECUTEBUFFERDESC “,,ffff ffff,*,,,,lpDatafff,, ffff  
fff,ffff,,Ž“f f,,ffff, ’,,, ,,f f, ffff,fff fff,‘ , dwCapsfff,,  
,Ž,,, ffff fff, IDirect3DExecuteBuffer::Lockffff,Ž,,Direct3D,ffff,  
fff fffŠ,,f f“,—<,,
```

Direct3DExecuteBuffer ののの IDirect3DDevice::Execute

*IDirect3DExecuteBuffer::Unlock***IDirect3DExecuteBuffer::Optimize**

```
HRESULT Optimize();
```

IDirect3DExecuteBuffer::SetExecuteData

```
HRESULT SetExecuteData(LPD3DEXECUTEDATA lpData);
```

Direct3DExecuteBuffer の Direct3DExecuteBuffer の

- `D3D_OK`, `D3DERR_EXECUTE_LOCKED`, `D3DERR_INVALIDOBJECT`, `D3DERR_INVALIDPARAMS`

lpData
の `D3DEXECUTEDATA` の

Direct3DExecuteBuffer の

IDirect3DExecuteBuffer::GetExecuteData

IDirect3DExecuteBuffer::Unlock

`HRESULT Unlock();`

の の `IDirect3DDevice::Execute`

- `D3D_OK`, `D3DERR_EXECUTE_NOT_LOCKED`, `D3DERR_INVALIDOBJECT`

IDirect3DExecuteBuffer::Lock

IDirect3DExecuteBuffer::Validate

`HRESULT Validate(LPDWORD lpdwOffset, LPD3DVALIDATECALLBACK lpFunc, LPVOID lpUserArg, DWORD dwReserved);`

IDirect3DLight

`IDirect3DLight` の の `IDirect3DLight` の *IDirect3DLight*

`IDirect3DLight` の の

`GetLight`
`SetLight`

`Initialize`

IDirect3DMaterial

IDirect3DMaterial の **IDirect3Dmaterial**
IDirect3DMaterial

IDirect3DMaterial の

| | |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>—</p> <p>ffff</p> <p>„‘</p> | <p>Reserve</p> <p>Unreserve</p> <p>GetMaterial</p> <p>SetMaterial</p> <p>GetHandle</p> <p>Initialize</p> |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|

IDirect3DMaterial の COM の
IUnknown の **IUnknown** の

AddRef
QueryInterface
Release

IDirect3DMaterial::GetHandle

```
HRESULT GetHandle(LPDIRECT3DDEVICE lpDirect3DDevice,
                  LPD3DMATERIALHANDLE lpHandle);
```

Direct3DMaterial の の の Direct3D API の の 1

- **DDERR_INVALIDOBJECT**, **DDERR_D3D_OK**, **DDERR_INVALIDPARAMS**

lpDirect3DDevice

Direct3DDevice ffffff, ffffff, ffffff, ffffff

lpHandle

Direct3DMaterial ffffff, '%', ffffff, ffffff, ' ', ffffff

IDirect3DMaterial::GetMaterial

```
HRESULT GetMaterial(LPD3DMATERIAL lpMat);
```

Direct3DMaterial の

- **DDERR_INVALIDOBJECT**, **DDERR_D3D_OK**, **DDERR_INVALIDPARAMS**

lpMat
 ⊕ ,fffff,“ , ’,,D3DMATERIAL “,,ffff
IDirect3DMaterial::SetMaterial

IDirect3DMaterial::Initialize

HRESULT Initialize(LPDIRECT3D lpDirect3D);

COM の の の

- Direct3DMaterialffffff, Ž, Š%
 ,,, DDERR_ALREADYINITIALIZED,•

lpDirect3D
 Direct3Dffffff,Ž,Direct3D “,,ffff

IDirect3DMaterial::Reserve

HRESULT Reserve();

IDirect3DMaterial::SetMaterial

HRESULT SetMaterial(LPD3DMATERIAL lpMat);

Direct3DMaterial の

- ⊕,,D3D_OK Ž”,, Ž’,,,,,,•,
 DDERR_INVALIDOBJECT
 DDERR_INVALIDPARAMS

lpMat
 ffffff“ , ’,,D3DMATERIAL “,,ffff
IDirect3DMaterial::GetMaterial

IDirect3DMaterial::Unreserve

HRESULT Unreserve();

IDirect3DTexture

IDirect3DTexture の の の の **IDirect3Dtexture** の **IDirect3DTexture**

IDirect3DTexture の の

ffff

GetHandle

Š%

Initialize

f f

Load

Unload

ffff •

PaletteChanged

IDirect3DTexture

IUnknown

の COM

IUnknown

の

AddRef

QueryInterface

Release

Direct3DTexture

DirectDrawSurface

IDirect3D::QueryInterface

IDirect3DTexture::GetHandle

```
HRESULT GetHandle(LPDIRECT3DDEVICE lpDirect3DDevice,
                  LPD3DTEXTUREHANDLE lpHandle);
```

Direct3DTexture のののの Direct3D API の

- **D3DERR_BADOBJECT**
DDERR_INVALIDPARAMS

lpDirect3DDevice

f f,,,ffff,Direct3DDeviceffffff,ffff

lpHandle

Direct3DTextureffffff,‘%,,ffff,ffff, ’,,• ,ffff

IDirect3DTexture::Initialize

```
HRESULT Initialize(LPDIRECT3DDEVICE lpD3DDevice,
                  LPDIRECTDRAWSURFACE lpDDSurface);
```

COM ののの

- **DDERR_ALREADYINITIALIZED**,•

lpDirect3D
Direct3Dfffff,Ž,Direct3D ‘,ffff

lpDDSurface
,fffff ,DirectDraw,,fff

IDirect3DTexture::Load

HRESULT Load(LPDIRECT3DTEXTURE lpD3DTexture);

DDSCAPS_ALLOCONLOAD

- Ą,, D3D_OK,•, ,,‘,,ff ,, •’,,,, Direct3D ’ f f,•’ ,Ž ,,

lpD3DTexture
f f,,fffff,,fff
IDirect3DTexture::Unload

IDirect3DTexture::PaletteChanged

HRESULT PaletteChanged(DWORD dwStart, DWORD dwCount);

Ń

- Ą,, D3D_OK,•, ,,‘,,ff ,, •’,,,, Direct3D ’ f f,•’ ,Ž ,,

dwStart
• ,, ‘,ffff,ffffff

dwCount
• ,,ffff,Ą

ŃŃŃŃ

IDirect3DTexture::Unload

HRESULT Unload();

Ń

- Ą,,D3D_OK,•, ,,‘,,ff ,, •’,,,, Direct3D ’ f f,•’ ,Ž ,,

IDirect3DTexture::Load

IDirect3DViewport

IDirect3DViewport ŃŃŃŃIDirect3DViewport ŃIDirect3DViewport

IDirect3DViewport ŃŃ

LPBOOL lpValid);

DirectDraw

- D3D_OK, DDERR_INVALIDOBJECT, DDERR_INVALIDPARAMS

lpDDSurface

DirectDrawSurface, ffffffff, ffff, ffff

lpValid

FALSE, ffffffff

IDirect3DViewport::SetBackgroundDepth

IDirect3DViewport::GetViewport

HRESULT GetViewport(LPD3DVIEWPORT lpData);

- D3D_OK, DDERR_INVALIDOBJECT, DDERR_INVALIDPARAMS

lpData

LPD3DVIEWPORT, ffffffff

IDirect3DViewport::SetViewport

IDirect3DViewport::Initialize

HRESULT Initialize(LPDIRECT3D lpDirect3D);

COM の の の

- Direct3DViewport, DDERR_ALREADYINITIALIZED,

lpDirect3D

LPDIRECT3D, ffffffff, LPDIRECT3D, ffffffff

IDirect3DViewport::LightElements

HRESULT LightElements(DWORD dwElementCount, LPD3DLIGHTDATA lpData);

lpDDSurface

”E “,DirectDrawSurfaceffffff,,fff

IDirect3DViewport::Clear

D3DCLEAR_ZBUFFER

Z

16

IDirect3DViewport::GetBackgroundDepth

IDirect3DViewport::SetViewport

HRESULT SetViewport(LPD3DVIEWPORT lpData);

- E,,D3D_OK Ž”,, Ž,’,,,,,*,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpData

<ff f f,Ž,D3DVIEWPORT “,,fff

IDirect3DViewport::GetViewport

IDirect3DViewport::TransformVertices

HRESULT TransformVertices(DWORD dwVertexCount,

LPD3DTRANSFORMDATA lpData, DWORD dwFlags, LPDWORD lpOffscreen);

- E,,D3D_OK Ž”,, Ž,’,,,,,*,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

dwVertexCount

itfff f,’,Š,,,”

lpData

•Š,,,”,Š,D3DTRANSFORMDATA “,,fff

dwFlags

^%o,fff,“,,, ,fff,Ž,*,,,,, E , -,Ž ,,,

D3DTRANSFORM_CLIPPED

D3DTRANSFORM_UNCLIPPED

lpOffscreen

•ŠE “,fff fŠ,, 0Š,’,’,,* ,,fff

it D3DTRANSFORM_CLIPPED のののののit

D3DLVERTEX のののit *it* 0 のD3DTRANSFORMDATA の

drExtent のの 2D

it D3DTRANSFORM_UNCLIPPED のののののののの
D3DTRANSFORMDATA の drExtent のの

D3DTRANSFORMDATA の dwClip のののののののの



D3DBRANCH

```
typedef struct _D3DBRANCH {
    DWORD dwMask;
    DWORD dwValue;
    BOOL bNegate;
    DWORD dwOffset;
} D3DBRANCH, *LPD3DBRANCH;
```

ののの

dwMask

•Š,,,ffffff ,,fff, ~— —,,,ffff ‘fff,☉ ,, ☉%
,dwValuefff,Ž’,,,’,, bNegatefff,FALSE, •Š,,
Ž—%”,ffff ‘fff,fff, D3DSTATUS “,dwStatus fff,Ž ,,,,

dwValue

dwMaskfff,Ž,,,—,”Š,,,ffff fff’<,’

bNegate

TRUE, ”Š☉%,”“

dwOffset

•Š ,,ffffff —,,, 0,Ž’,,

D3DCOLORVALUE

```
typedef struct _D3DCOLORVALUE {
    union {
        D3DVALUE r;
        D3DVALUE dvR;
    };
    union {
        D3DVALUE g;
        D3DVALUE dvG;
    };
    union {
        D3DVALUE b;
        D3DVALUE dvB;
    };
    union {
        D3DVALUE a;
        D3DVALUE dvA;
    };
} D3DCOLORVALUE;
```

D3DLIGHT D3DMATERIAL

dwR, dwG, dwB, dwA
 ,—',,, — ffff,Ž',,D3DVALUECE,'

D3DDEVICEDESC

```
typedef struct _D3DDeviceDesc {
    DWORD          dwSize;
    DWORD          dwFlags;
    D3DCOLORMODEL dcmColorModel;
    DWORD          dwDevCaps;
    D3DTRANSFORMCAPS dtcTransformCaps;
    BOOL           bClipping;
    D3DLIGHTINGCAPS dlcLightingCaps;
    D3DPRIMCAPS    dpcLineCaps;
    D3DPRIMCAPS    dpcTriCaps;
    DWORD          dwDeviceRenderBitDepth;
    DWORD          dwDeviceZBufferBitDepth;
    DWORD          dwMaxBufferSize;
    DWORD          dwMaxVertexCount;
} D3DDEVICEDESC, *LPD3DDEVICEDESC;
```

のののIDirect3DDevice::GetCaps のの

dwSize

,, “,fff”,fff

dwFlags

,, “,—CE,f f, ',,,,fff,Ž•,fff

D3DDD_BCLIPPING

bClippingfff,—CE

D3DDD_COLORMODEL

dcmColorModelfff,—CE

D3DDD_DEVCAPS

dwDevCapsfff,—CE

D3DDD_LIGHTINGCAPS

dlcLightingCapsfff,—CE

D3DDD_LINECAPS

dpcLineCapsfff,—CE

D3DDD_MAXBUFFERSIZE

dwMaxBufferSizefff,—CE

D3DDD_MAXVERTEXCOUNT

dwMaxVertexCountfff,—CE

D3DDD_TRANSFORMCAPS

dtcTransformCapsfff,—CE

D3DDD_TRICAPS

dpcTriCapsfff,—CE

dcmColorModel

ffff,ff fff,Ž',,,,D3DCOLORMODEL—(€fff,,,,

dwDevCaps

ffff,“—,Ž•,,fff

D3DDEVCAPS_EXECUTESYSTEMMEMORY

ffff fff,Ž ffff,Ž—%”

D3DDEVCAPS_EXECUTEVIDEOMEMORY

fff fff,Ž ffff,Ž—%”

D3DDEVCAPS_FLOATTLVERTEX

•Š€,”“ f f,“ “,Ž“

D3DDEVCAPS_SORTDECREASINGZ

“,€ •€ ,f f,,f f,•—

D3DDEVCAPS_SORTEXACT

-,f f,,f f,•—

D3DDEVCAPS_SORTINCREASINGZ

Z’%o•€ ,f f,,f f,•—

D3DDEVCAPS_TEXTURESYSTEMMEMORY

ffff fff,,fffff,Ž“%”

D3DDEVCAPS_TEXTUREVIDEOMEMORY

fffffff,,fffff,Ž“%”

D3DDEVCAPS_TLVERTEXSYSTEMMEMORY

“,•Š,,,–,,,,, ffff fff,ffff,Ž—%”

D3DDEVCAPS_TLVERTEXVIDEOMEMORY

“,•Š,,,–,,,,, fff fff,ffff,Ž—%”

dctTransformCaps

ffff,•Š—,Ž’,,D3DTRANSFORMCAPS “,fff,,,

bClipping

ffff,3Dfffff,Ž %”,,, TRUE,,,

dlcLightingCaps

ffff,€€—,Ž’,,D3DLIGHTINGCAPS “,fff,,,

dpcLineCaps□dpcTriCaps

,ŽŠ€,%o,,, ffff,ff f •,’<,D3DPRIMCAPS “

dwDeviceRenderBitDepth

ffff,fffff¥fff “ Ž,DirectDrawfff “ DDBD_8 DDBD_16 DBD_24 DDBD_32,“,,^ , ’,,

dwDeviceZBufferBitDepth

ffff,Zffff,fff “ Ž,DirectDrawfff “ DDBD_8 DDBD_16 DDBD_24 DDBD_32,“,,^ , ’,,

dwMaxBufferSize

,ffff,Ž ffff, ‘fff ,fff,0,,, ffff fff,,,,,fff,Ž—%”,,,

dwMaxVertexCount

,ffff, “

**D3DCOLORMODEL D3DFINDDEVICERESULT
D3DLIGHTINGCAPS D3DPRIMCAPS D3DTRANSFORMCAPS**

D3DEXECUTEBUFFERDESC

```
typedef struct _D3DExecuteBufferDesc {
    DWORD dwSize;
    DWORD dwFlags;
    DWORD dwCaps;
    DWORD dwBufferSize;
    LPVOID lpData;
} D3DEXECUTEBUFFERDESC;
typedef D3DEXECUTEBUFFERDESC *LPD3DEXECUTEBUFFERDESC;
```

IDirect3DDevice::CreateExecuteBuffer **IDirect3DExecuteBuffer::Lock**
①

dwSize

,, “,fff^,fff

dwFlags

,, “,—E,f f, ’,,,,,fff,Ž•,,fff

D3DDEB_BUFSIZE

dwBufferSizefff,—E

D3DDEB_CAPS

dwCapsfff,—E

D3DDEB_LPDATA

lpDatafff,—E

dwCaps

Ž ffff,fff,^’

D3DDEBCAPS_MEM

D3DDEBCAPS_SYSTEMMEMORY,

D3DDEBCAPS_VIDEOMEMORY,~—(OR)

D3DDEBCAPS_SYSTEMMEMORY

Ž ffff,f f,ffff fff,’ ,,

D3DDEBCAPS_VIDEOMEMORY

Ž ffff,f f,ffffff,’ ,,

dwBufferSize

Ž ffff,fff^,fff

lpData

ffff f f,,ffff

D3DEXECUTEDATA

```
typedef struct _D3DEXECUTEDATA {
    DWORD dwSize;
    DWORD dwVertexOffset;
    DWORD dwVertexCount;
    DWORD dwInstructionOffset;
    DWORD dwInstructionLength;
    DWORD dwHVertexOffset;
```

```

    D3DSTATUS dsStatus;
} D3DEXECUTEDATA, *LPD3DEXECUTEDATA;

```

IDirect3DDevice::Execute の **dwInstructionOffset**

dwSize

,, “,fff”^,fff

dwVertexOffset

”“fff,fffff

dwVertexCount

Ž ,,“

dwInstructionOffset

Ž —fff,fffff

dwInstructionLength

Ž —,’,

dwHVertexOffset

ffff fff,ffffff,•—,,fff f •f f,—,,,,,,Ž—,,“Ž,”“,fffff

dsStatus

•Š —ŒŽ—,,fff f”^,Š”,,’ ,,’, D3DSTATUS “,,,

D3DSTATUS

D3DFINDDEVICERESULT

```

typedef struct _D3DFINDDEVICERESULT {
    DWORD        dwSize;
    GUID         guid;
    D3DDEVICEDESC ddHwDesc;
    D3DDEVICEDESC ddSwDesc;
} D3DFINDDEVICERESULT, *LPD3DFINDDEVICERESULT;

```

IDirect3D::FindDevice

dwSize

,, “,fff”^,fff

guid

Œ ,,ffff,ff ff ff fŽ•Ž(GUID)

ddHwDesc **ddSwDesc**

Œ ,,f ffff,ffffff,ffff,Ž,D3DDEVICEDESC “

D3DFINDDEVICESEARCH

D3DFINDDEVICESEARCH

```

typedef struct _D3DFINDDEVICESEARCH {
    DWORD        dwSize;
    DWORD        dwFlags;
    BOOL         bHardware;
}

```

```

D3DCOLORMODEL dcmColorModel;
GUID          guid;
DWORD         dwCaps;
D3DPRIMCAPS  dpcPrimCaps;
} D3DFINDDEVICESEARCH, *LPD3DFINDDEVICESEARCH;

```

のIDirect3D::FindDevice

dwSize

```
,, “,fff”,fff
```

dwFlags

```
ffff fff,☉,,,ffff☉,’☉,fff,,fff,, Ž,’“,”^ , ,’,,,
```

D3DFDS_ALPHACMPCAPS

```
,, “,dpcPrimCapsfff,,Ž’,,,D3DPRIMCAPS “,dwAlphaCmpCap  
sfff,^’,,
```

D3DFDS_COLORMODEL

```
,, “,dcmColorModelfff,Ž’,,,ff ¥fff,^’,,
```

D3DFDS_DSTBLENDCAPS

```
,, “,dpcPrimCapsfff,,Ž’,,,D3DPRIMCAPS “,dwDestBlendCaps  
fff,^’,,
```

D3DFDS_GUID

```
,, “,guidfff,Ž’,,,ff ff ff fŽ•Ž(GUID),^’,,
```

D3DFDS_HARDWARE

```
,, “,bHardwarefff,—,,,,,f ffff,ffffff,Ž“—,^’,,
```

D3DFDS_LINES

```
D3DDEVICEDESC “,dpcLineCapsfff,Ž’,,, D3DPRIMCAPS “,^’  
”
```

D3DFDS_MISCCAPS

```
,, “,dpcPrimCapsfff,,Ž’,,,D3DPRIMCAPS “,dwMiscCapsfff,^’  
”
```

D3DFDS_RASTERCAPS

```
,, “,dpcPrimCapsfff,,Ž’,,,D3DPRIMCAPS “,dwRasterCapsfff  
,^’,,
```

D3DFDS_SHADECAPS

```
,, “,dpcPrimCaps  
fff,,Ž’,,,D3DPRIMCAPS “,dwShadeCapsfff,^’,,
```

D3DFDS_SRCBLENDCAPS

```
,, “,dpcPrimCapsfff,,Ž’,,,D3DPRIMCAPS “,dwSrcBlendCapsf  
ff,^’,,
```

D3DFDS_TEXTUREBLENDCAPS

```
,, “,dpcPrimCapsfff,,Ž’,,,D3DPRIMCAPS “, dwTextureBlend  
Capsfff,^’,,
```

D3DFDS_TEXTURECAPS

```
,, “,dpcPrimCaps  
fff,,Ž’,,, D3DPRIMCAPS “,dwTextureCapsfff,^’,,
```

D3DFDS_TEXTUREFILTERCAPS

```

,, “,dpcPrimCapsfff,,Ž’,,dwTextureCaps
“ , dwTextureFilterCaps fff,^’,,

```

D3DFDS_TRIANGLES**D3DDEVICEDESC**

```

“ , dwTextureFilterCapsfff,Ž’,, D3DPRIMCAPS “,^’,,

```

D3DFDS_ZCMPCAPS

```

,, “,dpcPrimCapsfff,,Ž’,,D3DPRIMCAPS “,dwZCmpCapsfff
,^’,,

```

bHardware

```

Ž“,,ffff, f ffff,ffffff,,,,,,,,,Ž’,,fff ,,fff,TRUE,,,, f ffff,fff,
Ž“,, ,, f ffff fffff fff, , ,fff,Ž—
,,ffff fff, dwFlagsfff,D3DFDS_HARDWAREfff, ’,,,

```

dcmColorModel

```

fff,RGBff ¥fff,Ž“,,ffff,Ž’,,D3DCOLORMODEL—(€fff,,,

```

guid

```

€ ,,ffff,ff ff ff fŽ•Ž(GUID)

```

dwCaps

```

”—,fff

```

dpcPrimCaps

```

,,,,,ffffff€,,,,ffff,—,Ž,D3DPRIMCAPS “,Ž’,,

```

D3DFINDDEVICERESULT**D3DHVERTEX**

```

typedef struct _D3DHVERTEX {
    DWORD        dwFlags;
    union {
        D3DVALUE hx;
        D3DVALUE dvHX;
    };
    union {
        D3DVALUE hy;
        D3DVALUE dvHY;
    };
    union {
        D3DVALUE hz;
        D3DVALUE dvHZ;
    };
} D3DHVERTEX, *LPD3DHVERTEX;

```

⓪D3DTRANSFORMDATA ⓪**dwFlags**

```

“Ž“,,ffffff ‘,’<,,fff ,,fff, D3DTRANSFORMDATA “,dwClipfff,
fff,,,^ ‘’,,

```

dvHX dvHY dvHZ

```

“Ž •,,•Š,Ž’,,D3DVALUE€,’ ,,,, •, “,Ž,

```

D3DINSTRUCTION

```
typedef struct _D3DINSTRUCTION {
    BYTE bOpcode;
    BYTE bSize;
    WORD wCount;
} D3DINSTRUCTION, *LPD3DINSTRUCTION;
```

ののののの

bOpcode

D3DOPCODE —(E,fff,,Ž',,,,ffffff—

bSize

—f fffff,fff ,,fff, Ž,Ž,—,”,,,Ž,,

wCount

’,,—f f ,,fff, ŽŠE,ffff,,,,,ŽŠE,,,, Ž,—,,,,Ž,Ž,,

D3DLIGHT

```
typedef struct _D3DLIGHT {
    DWORD dwSize;
    D3DLIGHTTYPE dltType;
    D3DCOLORVALUE dcvColor;
    D3DVECTOR dvPosition;
    D3DVECTOR dvDirection;
    D3DVALUE dvRange;
    D3DVALUE dvFalloff;
    D3DVALUE dvAttenuation0;
    D3DVALUE dvAttenuation1;
    D3DVALUE dvAttenuation2;
    D3DVALUE dvTheta;
    D3DVALUE dvPhi;
} D3DLIGHT, *LPD3DLIGHT;
```

IDirect3DLight::SetLight IDirect3DLight::GetLight のの

dwSize

,, “,fff”^,fff

dltType

(E,E ,,’, D3DLIGHTTYPE—(E,fff,,,,,,

dcvColor

(E,E, ,,fff, D3DCOLORVALUE “,,

dvPosition□dvDirection

f ff<Š,,(E,E,^,”^

dvRange

(E,E,—E”^

dvFalloff

%o(dvThetafff,Ž’,,Š“),”%o,Š~(dvPhifff,Ž’,,Š“)
,Š,<“,(E ,, ”%o“,,<“ Ž,•Ž,Ž,,

D3DLIGHTINGCAPS

```
typedef struct _D3DLIGHTINGCAPS {
    DWORD dwSize;
    DWORD dwCaps;
    DWORD dwLightingModel;
    DWORD dwNumLights;
} D3DLIGHTINGCAPS, *LPD3DLIGHTINGCAPS;
```

のD3DDEVICEDESCの

dwSize

、 “,fff”^,fff

dwCaps

☒☒fff f,”—,Ž,fff Ž,fff,’<,,,,

D3DLIGHTCAPS_DIRECTIONAL

’Ž☒,ff f,,

D3DLIGHTCAPS_GLSPOT

Open-GLffff,ffffff,ff f,,

D3DLIGHTCAPS_PARALLELPOINT

• ☒,ff f,,

D3DLIGHTCAPS_POINT

“☒☒,ff f,,

D3DLIGHTCAPS_SPOT

ffffff,ff f,,

dwLightingModel

☒☒fff, RGB,ffff,,,,,’<,fff Ž,fff,’<,,,,

D3DLIGHTINGMODEL_MONO ffff☒☒fff

D3DLIGHTINGMODEL_RGB RGB☒☒fff

dwNumLights

Ž—,,,☒☒

D3DLIGHTINGELEMENT

```
typedef struct _D3DLIGHTINGELEMENT {
    D3DVECTOR dvPosition;
    D3DVECTOR dvNormal;
} D3DLIGHTINGELEMENT, *LPD3DLIGHTINGELEMENT;
```

のD3DLIGHTDATAの

dvPosition

fff<Š,, -^,Ž’,,’ ,,’, D3DVECTOR “,,,

dvNormal

- ffff,Ž’,,’ ,,’, D3DVECTOR “,,,

D3DLIGHTDATA IDirect3DViewport::LightElements

D3DLINE

```
typedef struct _D3DLINE {
    union {
        WORD v1;
        WORD wV1;
    };
    union {
        WORD v2;
        WORD wV2;
    };
} D3DLINE, *LPD3DLINE;
```

D3DOPCODE の D3DOP_LINE の

wV1 wV2

“,ffffff

1

D3DLINEPATTERN

```
typedef struct _D3DLINEPATTERN {
    WORD wRepeatFactor;
    WORD wLinePattern;
} D3DLINEPATTERN;
```

の D3DRENDERSTATETYPE の D3DRENDERSTATE_LINEPATTERN

wRepeatFactor

ff fŠŽ, ”,,, ŠŽ‘,Ž—,,,wLinePatternfff,Ž’,,,ff f,fff

wLinePattern

fff¥fff f,Ž’,,,fff ,,,, ’1100110011001100, fff¥fff, ,,

D3DLVERTEX

```
typedef struct _D3DLVERTEX {
    union {
        D3DVALUE x;
        D3DVALUE dvX;
    };
    union {
        D3DVALUE y;
        D3DVALUE dvY;
    };
    union {
        D3DVALUE z;
        D3DVALUE dvZ;
    };
    DWORD dwReserved;
    union {
        D3DCOLOR color;
```



```
} D3DMATERIAL, *LPD3DMATERIAL;
```

IDirect3DMaterial::GetMaterial **IDirect3DMaterial::SetMaterial** ののの

dwSize

,, “,fff”^,fff

d3vDiffuse d3vAmbient d3vSpecular d3vEmissive

,,, ,ffff,ŠŽ • ^ “%o •Ž ,Ž’,,’ ,,,’, D3DCOLORVALUE “,

”

d3vPower

“%o,ffff,č,Ž’,,D3DVALUEč,’

hTexture

fffff¥fff,ffff

dwRampSize

,čŽ,’,’ ,(čŽ)ffff,, ”čŽ’,,,,fffff,.,,’ ,1^%o

,,,,,,,,,,’,’,’ ”čŽ’,,.,,’ ”čŽ’,,fffff,dwRampSizefff,1^ ,fff

ff,č ,,,,,,,,,,’,’

ののの

IDirect3DMaterial::GetMaterial *IDirect3DMaterial::SetMaterial*

D3DMATRIX

```
typedef struct _D3DMATRIX {
    D3DVALUE _11, _12, _13, _14;
    D3DVALUE _21, _22, _23, _24;
    D3DVALUE _31, _32, _33, _34;
    D3DVALUE _41, _42, _43, _44;
} D3DMATRIX, *LPD3DMATRIX;
```

IDirect3DDevice::GetMatrix **IDirect3DDevice::SetMatrix**

Direct3D の_44 のの-1

IDirect3DDevice::GetMatrix *IDirect3DDevice::SetMatrix*

D3DMATRIXLOAD

```
typedef struct _D3DMATRIXLOAD {
    D3DMATRIXHANDLE hDestMatrix;
    D3DMATRIXHANDLE hSrcMatrix;
} D3DMATRIXLOAD, *LPD3DMATRIXLOAD;
```

D3DOPCODE の **D3DOP_MATRIXLOAD** のの¥

hDestMatrix □ **hSrcMatrix**

“ ,“č, —,fff

D3DOPCODE

D3DMATRIXMULTIPLY

```
typedef struct _D3DMATRIXMULTIPLY {
    D3DMATRIXHANDLE hDestMatrix;
    D3DMATRIXHANDLE hSrcMatrix1;
    D3DMATRIXHANDLE hSrcMatrix2;
} D3DMATRIXMULTIPLY, *LPD3DMATRIXMULTIPLY;
```

D3DOPCODE の **D3DOP_MATRIXMULTIPLY** のの¥

hDestMatrix

“Ž“%o,Š”,, —,ffff

hSrcMatrix1 □ **hSrcMatrix2**

1”-,2”-,“Ž‘ , —,ffff

D3DOPCODE

D3DPICKRECORD

```
typedef struct _D3DPICKRECORD {
    BYTE    bOpcode;
    BYTE    bPad;
    DWORD   dwOffset;
    D3DVALUE dvZ;
} D3DPICKRECORD, *LPD3DPICKRECORD;
```

IDirect3DDevice::GetPickRecords のの

bOpcode

“ffffff,fff f

bPad

Padfff

dwOffset

“ ,,“ffffff,Ž ffff, “,,fffff

dvZ

“ffffff, “

の x y

IDirect3DDevice::Pick

IDirect3DDevice::GetPickRecords *IDirect3DDevice::Pick*

D3DPOINT

```
typedef struct _D3DPOINT {
    WORD wCount;
    WORD wFirst;
} D3DPOINT, *LPD3DPOINT;
```

D3DOPCODE の **D3DOP_POINT** のの¥

wCount

“,

wFirst
 ,’“,ffffff

D3DOPCODE

D3DPRIMCAPS

```
typedef struct _D3DPrimCaps {
    DWORD dwSize;
    DWORD dwMiscCaps;
    DWORD dwRasterCaps;
    DWORD dwZCmpCaps;
    DWORD dwSrcBlendCaps;
    DWORD dwDestBlendCaps;
    DWORD dwAlphaCmpCaps;
    DWORD dwShadeCaps;
    DWORD dwTextureCaps;
    DWORD dwTextureFilterCaps;
    DWORD dwTextureBlendCaps;
    DWORD dwTextureAddressCaps;
    DWORD dwStippleWidth;
    DWORD dwStippleHeight;
} D3DPRIMCAPS, *LPD3DPRIMCAPS;
```

○○○○○D3DDEVICEDESC ○○○

dwSize
 ,, “,fff”^,fff

dwMiscCaps
 ,,ffffff,““,”— ,,fff,, Ž’,,,,^ , ’,,

D3DPMISCCAPS_CONFORMANT

ffff, OpenGL• , ,,,

D3DPMISCCAPS_CULLCCW

ffff, D3DRENDERSTATE_CULLMODE ‘,“, %
 ,,ff f,,(ŽŠŒ,ffffff, ,,) D3DCULL
 —<Œ,D3DCULL_CCWfff,%o,,

D3DPMISCCAPS_CULLCW

ffff, D3DRENDERSTATE_CULLMODE ‘,“,,%o%
 ,,ŽŠŒ,ff f,,(ŽŠŒ,ffffff, ,,) D3DCULL
 —<Œ,D3DCULL_CCWfff,%o,,

D3DPMISCCAPS_CULLNONE

ffff, “,,,ŽŠŒ,Ž ,,,, D3DCULL—<Œ,D3DCULL_NONEfff,%o
 ”

D3DPMISCCAPS_LINEPATTERNREP

ffff, D3DLINEPATTERN “,wRepeatFactorfff,1^ ,’,Ž—
 ,,(' *%o,ffffff, ,,)

D3DPMISCCAPS_MASKPLANES

ffff, ,ffffff,Ž ,,,

D3DPMISCCAPS_MASKZ

ffff, ffff —,Zffff, ,%”,,,,•%”,,,,,,

dwRasterCaps

fff•%”,— • ,fff,, Ž’,,,,^ ’,,,

D3DPRASTERCAPS_DITHER

ffff, , Ć—,—,,,,,fff•Ž’,,,

D3DPRASTERCAPS_FOGTABLE

ffff, ffff, ,,Ž,fff’,•Š,,Ž“f ff, ,, fff’,Ž ,,

D3DPRASTERCAPS_FOGVERTEX

ffff, — —,Š’ 3DTLVERTEX “,specularfff,—
,,,D3DCOLOR’,ffff •’,’ĆŽ,, ,, fff,Š’ fff’, ,Š,,

D3DPRASTERCAPS_PAT

ffff, ff f•%”, —,, (D3DRENDERSTATE_LINEPATTERN,
D3DRENDERSTATE_STIPPLEPATTERNffffff ‘,,,,, •,“,,,,)ffff
ff—, %”,,,

D3DPRASTERCAPS_ROP2

ffff, R2_COPYPEN^Š,fff —,ff f,,

D3DPRASTERCAPS_STIPPLE

ffff, ”“—,ffff f,,ffff,“”,,,

D3DPRASTERCAPS_SUBPIXEL

ffff, ^”<, Ć,ffff •,, Z fffff,f f,ffffff””, —
,, ,, Z,ff f ”,,,, ,(,fff,fff,ffffff’) ,,, —Ć,,,,—
Ć,,,,,,,(ffff, ffffff””,Ž ,, ,,,,,,,) ,,fff, ffffff,Ž,Ć’ ,,,,,,
Direct3Dffffff,’<,,,

D3DPRASTERCAPS_SUBPIXELX

ffff, XŽ,%”
,,ffffff””,,, Ć,YŽ,‘ , , ,,,,,, ffffff””,,, , D3DPRAS
TERCAPS_SUBPIXEL,Ž ,,,

D3DPRASTERCAPS_XOR

ffff, ”“~ — (XOR)
,ff f,, ,fff, ’,, D3DPRIM_RASTER_ROP2, ’,,,,, , ”“~—
~ —,ff f,,,,,,

D3DPRASTERCAPS_ZTEST

ffff,Zfff —
,Ž ,, ,, Ć%”,,ffffff,ffffff, ffffff,,,,,,Zffff,,•Ž,,

dwZCmpCaps

ffff,Ž %””,Š ,”Š,,Zffff ,fff, Ž’,,,,^ ’,,,

D3DPCMPCAPS_ALWAYS

,,Zfff,’

D3DPCMPCAPS_EQUAL

<,Z,Ć ,Z,“,, Zfff,’

D3DPCMPCAPS_GREATER
 $\langle Z, \mathcal{E}, Z, ', ', Zfff, ' \rangle$
D3DPCMPCAPS_GREATEREQUAL
 $\langle Z, \mathcal{E}, Z, ', ', ', ', Zfff, ' \rangle$
D3DPCMPCAPS_LESS
 $\langle Z, \mathcal{E}, Z, ', ', ', Zfff, ' \rangle$
D3DPCMPCAPS_LESSEQUAL
 $\langle Z, \mathcal{E}, Z, ', ', ', ', Zfff, ' \rangle$
D3DPCMPCAPS_NEVER
 $\langle ', Zfff, \checkmark \rangle$
D3DPCMPCAPS_NOTEQUAL
 $\langle Z, \mathcal{E}, Z, ', ', ', Zfff, ' \rangle$
dwSrcBlendCaps
 $f f \cdot ', fff, \checkmark, ', ', ^ ', ', (RGBA', f f, fffff fff, ', ', s, d, \bullet)$
D3DPBLENDCAPS_BOTHINVSRCALPHA
 $f f \text{ — '(1-As, 1-As, 1-As, 1-As), fffff fff — '(As, As, As, As) fffff fff ', -\mathcal{E}, ',$
D3DPBLENDCAPS_BOTHSRCALPHA
 $f f \text{ — '(As, As, As, As), fffff fff — '(1-As, 1-As, 1-As, 1-As) fffff fff ', -\mathcal{E}, ',$
D3DPBLENDCAPS_DESTALPHA
 $\text{— '(Ad, Ad, Ad, Ad)}$
D3DPBLENDCAPS_DESTCOLOR
 $\text{— '(Rd, Gd, Bd, Ad)}$
D3DPBLENDCAPS_INVDESTALPHA
 $\text{— '(1-Ad, 1-Ad, 1-Ad, 1-Ad)}$
D3DPBLENDCAPS_INVDESTCOLOR
 $\text{— '(1-Rd, 1-Gd, 1-Bd, 1-Ad)}$
D3DPBLENDCAPS_INVSRCALPHA
 $\text{— '(1-As, 1-As, 1-As, 1-As)}$
D3DPBLENDCAPS_INVSRCOLOR
 $\text{— '(1-Rd, 1-Gd, 1-Bd, 1-Ad)}$
D3DPBLENDCAPS_ONE
 — '(1, 1, 1, 1)
D3DPBLENDCAPS_SRCALPHA
 $\text{— '(As, As, As, As)}$
D3DPBLENDCAPS_SRCALPHASAT
 $\text{— '(f, f, f, 1) } f = \min(As, 1-Ad)$
D3DPBLENDCAPS_SRCCOLOR
 $\text{— '(Rs, Gs, Bs, As)}$

D3DPBLENDCAPS_ZERO

—(0, 0, 0, 0)

dwDestBlendCaps

ffff fff • ,fff, dwSrcBlendCapsfff,'<,,,"— •,",,,

dwAlphaCmpCaps

ffff,Ž %”,”Š,”Š,,ffff fff ,fff, dwZCmpCapsfff,'<,,,"—
•,",,,

dwShadeCaps

ffff,Ž ,,,ff ffff — ~”“, ffff,—,,,ffff(D3DOP_TRIANGLE,,,) ,Ž ,,, , D3DSHADE_FLATf f(D3DSHADEMODE—<E,Ž',,,,,) ,ff f,, ,fff, ff ff ffff,fff ff ffff,ff f,ffff, ,,,ff f f,,,,ff f,,ffff ,Ž',, ffff—‘,— ,,,f f,ff f,, , ,f f, , , ,ffff',•,255,, , , , ffff, “(ffff , <),,, ŽŠE, fffff“%o fff ffff •,,,,, ffff ffff,ffff fff,”,”— fff,, ,fff, ff ffff f f ff fff , , ,“,,, ,ffff • , , , , ffff ,Ž ,,, ,fff, Ž',,,,^ ,,,

D3DP SHADECAPS_ALPHAFLATBLEND

D3DP SHADECAPS_ALPHAFLATSTIPPLED

ffff, “%o“,,,, ,“•,,•-(D3DSHADEMODE —<E,D3DSHADE_FLAT) ,ffff •,ff f,, ,,,f f, ,,,—‘,,,,ffff • , —‘ , ,“ , ,1•,,,—,,,

D3DP SHADECAPS_ALPHAGOURAUDBLEND

D3DP SHADECAPS_ALPHAGOURAUDSTIPPLED

ffff, “%o“,,,, ,“•,,ff (D3DSHADEMODE —<E,D3DSHADE_GOURAUD) ,ffff •,ff f,, ,,,f f, ffff •,”,,,,, ‘ , •,,,, , , ,

D3DP SHADECAPS_ALPHAPHONGBLEND

D3DP SHADECAPS_ALPHAPHONGSTIPPLED

ffff, “%o“,,,, ,“•,,fff(D3DSHADEMODE —<E,D3DSHADE_PHONG),ffff •,ff f,, ,,,f f, “fff f, — , •,“%o,,E%E%o,ffff^ , •%o,,

D3DP SHADECAPS_COLORFLATMONO

D3DP SHADECAPS_COLORFLATRGB

ffff, D3DCOLOR_MONO,D3DCOLOR_RGBff f f,•- ff ffff, •,ff f,, ,,,f f, ,,,—‘ , • , —‘ , ,“ , , , , , ffffE%Ef f, , • , ,Š,,, RGBE%Ef f, — • , Š,,,

D3DP SHADECAPS_COLORGOURAUDMONO

D3DP SHADECAPS_COLORGOURAUDRGB

ffff, D3DCOLOR_MONO,D3DCOLOR_RGBff f f,ff ff f fff, •,ff f,, ,,,f f, ,,,—‘,,,,ffff •,”,,,,, ‘ , •,,,,•-

dwTextureFilterCaps

fffff,fffff • ,fff, Ž',,,,^ ',,,

D3DPTFILTERCAPS_LINEAR

ffff,Ž,Š,ffff,2 2—^, ,•,,,•' ,,, Š', ,—•,Ž—
,, Š', ,ff f,,,,,, —•,ff f,,,,,,

D3DPTFILTERCAPS_LINEARMIPLINEAR

D3DPRIM_TEX_MIP_LINEAR,“—,, ,,,,Žffffff, ,Š,,

D3DPTFILTERCAPS_LINEARMIPEAREST

D3DPRIM_TEX_MIP_NEAREST,“—,, ,,,,Žffffff, ,Š,,

D3DPTFILTERCAPS_MIPLINEAR

D3DPRIM_TEX_LINEAR,“—,, “ffff,‘%o,,ffffff,Ž—,,

D3DPTFILTERCAPS_MIPNEAREST

D3DPRIM_TEX_NEAREST,“—,, “ffff,‘%o,,ffffff,Ž—,,

D3DPTFILTERCAPS_NEAREST

ffff',Ž •,ffff,Ž—,, ,,, Š', ,—•,Ž—,, Š', ,ff f,,,,,,
—•,ff f,,,,,,

dwTextureBlendCaps

fffff • fffff f f,Ž',,,,D3DTEXTUREBLEND—<E, %o
,, ,fff, Ž',,,,^ ',,,

D3DPTBLENDCAPS_COPY

fffff f f,ff (D3DTEXTUREBLEND
—<E,,D3DPTBLEND_COPY),ff f,,

D3DPTBLENDCAPS_DECAL

ff f¥fffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_DECAL),ff f,,

D3DPTBLENDCAPS_DECALALPHA

ff f¥fffff¥fffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_DECALALPHA),ff f,,

D3DPTBLENDCAPS_DECALMASK

ff f¥fff¥fffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_DECALMASK),ff f,,

D3DPTBLENDCAPS_MODULATE

•'fffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_MODULATE),ff f,,

D3DPTBLENDCAPS_MODULATEALPHA

•'fffff¥fffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_MODULATEALPHA),ff f,,

D3DPTBLENDCAPS_MODULATEMASK

•'fff¥fffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_MODULATEMASK),ff f,,

dwTextureAddressCaps

fffff,fffff • ,fff, Ž',,,,^ ',,,

D3DPTADDRESSCAPS_CLAMP*ffff, fffff, Ć', %'***D3DPTADDRESSCAPS_MIRROR***ffff, fffff, ", %'***D3DPTADDRESSCAPS_WRAP***ffff, fffff, ', *, %'***dwStippleWidth** □ **dwStippleHeight***ff f, ", *, ', (32 32^)***D3DPROCESSVERTICES**

```
typedef struct _D3DPROCESSVERTICES {
    DWORD dwFlags;
    WORD wStart;
    WORD wDest;
    DWORD dwCount;
    DWORD dwReserved;
} D3DPROCESSVERTICES, *LPD3DPROCESSVERTICES;
```

D3DOPCODE の **D3DOP_PROCESSVERTICES****dwFlags***ffff, ", —, Ž', Ž, fff, , , ^ ', ,***D3DPROCESSVERTICES_COPY***" *, ffff, —, , Ž fff, ', " *, —, , , ffff, ", —, —
,,, , Ć% — " %', ,***D3DPROCESSVERTICES_NOCOLOR***" , * , , , ,***D3DPROCESSVERTICES_OPMASK****D3DPROCESSVERTICES_NOCOLOR,****D3DPROCESSVERTICES_UPDATEEXTENTS**, Ž , dwFlags fff, ', f
ff, fffff, Ž', ,**D3DPROCESSVERTICES_TRANSFORM***" , Š, , ,***D3DPROCESSVERTICES_TRANSFORMLIGHT***" , Š, , — Ć% , —, , , ,***D3DPROCESSVERTICES_UPDATEEXTENTS***, , , Š Ć, " , ^ , , , D3DSTATUS " , drExtent fff, *,***wStart***f f, , ", ffffff***wDest***f ff¥ffff, , ", ffffff***dwCount***—, , "*

dwReserved

— , 0,,,,,,,,,

D3DOPCODE

D3DRECT

```
typedef struct _D3DRECT {
    union {
        LONG x1;
        LONG lX1;
    };
    union {
        LONG y1;
        LONG lY1;
    };
    union {
        LONG x2;
        LONG lX2;
    };
    union {
        LONG y2;
        LONG lY2;
    };
} D3DRECT, *LPD3DRECT;
```

IX1 □IY1

<E, <, •

IX2 □IY2

<E,%o%o<, •

**D3DRMUPDATECALLBACK IDirect3DDevice::Pick
IDirect3DViewport::Clear**

D3DSPAN

```
typedef struct _D3DSPAN {
    WORD wCount;
    WORD wFirst;
} D3DSPAN, *LPD3DSPAN;
```

D3DOPCODE の D3DOP_SPAN の y

y

wCount

fff,

wFirst

,“,ffffff

D3DOPCODE


```

    D3DRECT drExtent;
} D3DSTATUS, *LPD3DSTATUS;

```

ののD3DEXECUTEDATA の D3DOPCODE の D3DOP_SETSTATUS

dwFlags

```

ff ff ^ ,,,—•,,,,,Ž',Ž,fff,,,, ' ,

```

D3DSETSTATUS_STATUS

```

ff ff, '

```

D3DSETSTATUS_EXTENTS

```

drExtentfff,Ž',,,^, '

```

D3DSETSTATUS_ALL

```

ff ff,^^,—•, '

```

dwStatus

```

ffffff¥fff ,fff, Ž,fff,,,^ , ',,

```

```

', ,,,^”fff

```

D3DSTATUS_CLIPINTERSECTION

```

,,,,CLIPINTERSECTIONfff,‘, ,,

```

D3DSTATUS_CLIPUNIONALL

```

,,,,CLIPUNIONfff,‘, ,,

```

D3DSTATUS_DEFAULT

```

D3DSTATUS_CLIPINTERSECTION,

```

```

D3DSTATUS_ZNOTVISIBLEfff,‘, ,, ,’, fffff,,

```

D3DSTATUS_ZNOTVISIBLE

```

ffff€ fff

```

D3DSTATUS_CLIPINTERSECTIONBACK

```

%o €€•ffff-,’“,ffffff fff,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONBOTTOM

```

%o €,%o,’“,ffffff fff,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONFRONT

```

%o €,’•ffff-,’“,ffffff fff,~— ,,,

```

**D3DSTATUS_CLIPINTERSECTIONGEN0 D3DSTATUS_CLIPINTERSECTI
ONGEN5**

```

ffff fff’<,ffff-~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONLEFT

```

%o €, ‘,’“,ffffff fff,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONRIGHT

```

%o €,%o,’“,ffffff fff,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONTOP

```

%o €, ,’“,ffffff fff,~— ,,,

```

```

ffffE fff
D3DSTATUS_CLIPUNIONBACK
    D3DCLIP_BACK,“,
D3DSTATUS_CLIPUNIONBOTTOM
    D3DCLIP_BOTTOM,“,
D3DSTATUS_CLIPUNIONFRONT
    D3DCLIP_FRONT,“,
D3DSTATUS_CLIPUNIONGEN0 D3DSTATUS_CLIPUNIONGEN5
    D3DCLIP_GEN0 D3DCLIP_GEN5,“,
D3DSTATUS_CLIPUNIONLEFT
    D3DCLIP_LEFT,“,
D3DSTATUS_CLIPUNIONRIGHT
    D3DCLIP_RIGHT,“,
D3DSTATUS_CLIPUNIONTOP
    D3DCLIP_TOP,“,

```

```

Š-ffffff¥fff
D3DCLIP_BACK
    %0 E,”-,,,ffffff,,
D3DCLIP_BOTTOM
    %0 E,’-,,,ffffff,,
D3DCLIP_FRONT
    %0 E,‘-,,,ffffff,,
D3DCLIP_LEFT
    %0 E, ‘-,,,ffffff,,
D3DCLIP_RIGHT
    %0 E,%0‘-,,,ffffff,,
D3DCLIP_TOP
    %0 E, ’-,,,ffffff,,
D3DCLIP_GEN0 D3DCLIP_GEN5
    ffff fff’<,ffff-

```

drExtent

```

,,,“,,Š,<E,’<,,D3DRECT “ ,,,, D3DPROCESSVERTICES “,’
’,,,D3DPROCESSVERTICES_UPDATEEXTENTSfff,Ž“,,D3DOP_PR
OCESSVERTICESfff f, —,Š,—^,’<,,

```

のD3DOP_SETSTATUS

D3DEXECUTEDATA D3DOPCODE D3DRECT

D3DTEXTURELOAD

```
typedef struct _D3DTEXTURELOAD {
```

```

        D3DTEXTUREHANDLE hDestTexture;
        D3DTEXTUREHANDLE hSrcTexture;
    } D3DTEXTURELOAD, *LPD3DTEXTURELOAD;

```

D3DOPCODE の D3DOP_TEXTURELOAD の

hDestTexture

“ fffff,ffff

hSrcTexture

“ fffff,ffff

hDestTexture hSrcTexture

D3DTLVERTEX

```

typedef struct _D3DTLVERTEX {
    union {
        D3DVALUE sx;
        D3DVALUE dvSX;
    };
    union {
        D3DVALUE sy;
        D3DVALUE dvSY;
    };
    union {
        D3DVALUE sz;
        D3DVALUE dvSZ;
    };
    union {
        D3DVALUE rhw;
        D3DVALUE dvRHW;
    };
    union {
        D3DCOLOR color;
        D3DCOLOR dcColor;
    };
    union {
        D3DCOLOR specular;
        D3DCOLOR dcSpecular;
    };
    union {
        D3DVALUE tu;
        D3DVALUE dvTU;
    };
    union {
        D3DVALUE tv;
        D3DVALUE dvTV;
    };
} D3DTLVERTEX, *LPD3DTLVERTEX;

```

D3DLIGHTDATA の(の)

dvSX, dvSY, dvSZ

fff f •, “, Ž’,, D3DVALUE(€, ’

dvRHW

D3DVALUE(, , , ,Z,%,,E“, ,ffffff,←,Š,’,,,

dcColor □ **dcSpecular**

“, ,%,Ž’,,D3DCOLOR(,’,

dvTU □ **dvTV**

“, ,ffffffŽ’,,D3DVALUE(,’,

D3DLIGHTDATA

D3DTRANSFORMCAPS

```
typedef struct _D3DTransformCaps {
    DWORD dwSize;
    DWORD dwCaps;
} D3DTRANSFORMCAPS, *LPD3DTRANSFORMCAPS;
```

のD3DDEVICEDESCの

dwSize

,, “,fff”^,fff

dwCaps

•Š’,ffff,ffffff,,,,,Ž’,,fff ,fff, 0,,Ž,fff, ’,,

D3DTRANSFORMCAPS_CLIP ffff,•Š’,ffffff,,

D3DTRANSFORMDATA

```
typedef struct _D3DTRANSFORMDATA {
    DWORD dwSize;
    LPVOID lpIn;
    DWORD dwInSize;
    LPVOID lpOut;
    DWORD dwOutSize;
    LPD3DHVERTEX lpHOut;
    DWORD dwClip;
    DWORD dwClipIntersection;
    DWORD dwClipUnion;
    D3DRECT drExtent;
} D3DTRANSFORMDATA, *LPD3DTRANSFORMDATA;
```

IDirect3DViewport::TransformVertices の**dwSize**

,, “,fff”^,fff

lpIn

•Š,,,’“,,ffff ,,, **D3DLVERTEX** “,’,,,

dwInSize

•Š,,,’“,,ŠŠ

lpOut

•Š,,,’“,,ffff

dwOutSize

—,ŠŠ

lpHOut

Ž•Š,,,’Š’,,ffff ,,’, D3DHVERTEX “,,

dwClip

’“,,,,,ffffff,,,,,Ž’,,fff ,,fff,, Ž’,,,,^ ’,,

D3DCLIP_BACK

%o E,’,,ffffff,,

D3DCLIP_BOTTOM

%o E,’-,,ffffff,,

D3DCLIP_FRONT

%o E,’-,,ffffff,,

D3DCLIP_GEN0 through D3DCLIP_GEN5

ffff fff’,ffff-

D3DCLIP_LEFT

%o E, ‘-,,ffffff,,

D3DCLIP_RIGHT

%o E,%o’-,,ffffff,,

D3DCLIP_TOP

%o E, ’-,,ffffff,,

dwClipIntersection

ffffff fff,E ,Ž,fff ,,fff,, Ž’,,,,^ ’,,

D3DSTATUS_CLIPINTERSECTIONBACK

%o E,E•ffff-,’,,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONBOTTOM

%o E,%o,’“,,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONFRONT

%o E,’•ffff-,’,,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONGEN0 D3DSTATUS_CLIPINTERSECTIONGEN5

ffff fff’,ffff-,~— ,,,

D3DSTATUS_CLIPINTERSECTIONLEFT

%o E, ‘’,,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONRIGHT

%o E,%o’,’,,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONTOP

%o E, ’’,,ffffff fff,~— ,,,

dwClipUnion

ffffff fff,E ,Ž,fff ,,fff,, Ž’,,,,^ ’,,

D3DSTATUS_CLIPUNIONBACK

```

D3DCLIP_BACK,“,
D3DSTATUS_CLIPUNIONBOTTOM
D3DCLIP_BOTTOM,“,
D3DSTATUS_CLIPUNIONFRONT
D3DCLIP_FRONT,“,
D3DSTATUS_CLIPUNIONGEN0 D3DSTATUS_CLIPUNIONGEN5
D3DCLIP_GEN0 D3DCLIP_GEN5,“,
D3DSTATUS_CLIPUNIONLEFT
D3DCLIP_LEFT,“,
D3DSTATUS_CLIPUNIONRIGHT
D3DCLIP_RIGHT,“,
D3DSTATUS_CLIPUNIONTOP
D3DCLIP_TOP,“,

```

drExtent

```

•ŠĀ,“;”^,Ā’,’,’, “, fffff•ŠĀ,fff f”^,•Šfff f,,,fff,,, ffffff,,,
, ,”^,Ž “;“,,,Š, ,’,D3DRECT “,,,

```

の[x y z]のdwInSize の

16

のの

IDirect3DViewport::TransformVertices

D3DTRIANGLE

```

typedef struct _D3DTRIANGLE {
    union {
        WORD v1;
        WORD wV1;
    };
    union {
        WORD v2;
        WORD wV2;
    };
    union {
        WORD v3;
        WORD wV3;
    };
    WORD wFlags;
} D3DTRIANGLE, *LPD3DTRIANGLE;

```

のの

D3DOPCODE のD3DOP_TRIANGLE

```

wV1 wV2 wV3
ŽŠĀ,Ž’,’,“

```

wFlags

Ž—%”,ŽŠĀ,fff,’,’,fff(,, •, fffff f f f,,—
(E) ,,fff,, Ž’,,,,^ ’,,

fff¥fff

D3DTRIFLAG_EDGEENABLE1

fffv1-v2,’,’,

D3DTRIFLAG_EDGEENABLE2

fffv2-v3,’,’,

D3DTRIFLAG_EDGEENABLE3

fffv3-v1,’,’,

D3DTRIFLAG_EDGEENABLETRIANGLE

,,,fff,’,’,

“Š , Āfff

D3DTRIFLAG_EVEN

Ā ,ŽŠĀ,v1-v2fff, ’’,ŽŠĀ,v3-
v1fff, ,,,, v1,’,v1, v2,’,v3,,,

D3DTRIFLAG_ODD

Ā ,ŽŠĀ,v1-v2fff, ’’,ŽŠĀ,v2-
v3fff, ,,,, v1,’,v3, v2,’,v2,,,

D3DTRIFLAG_START

“Š , Ā,ŠŽ,, ,,“f f,,

D3DTRIFLAG_STARTFLAT(len)

„ŽŠĀ, ,”,,,, ,Ā,ŽŠĀ,Ž’, , ,” , ,” , 0,,’,30,, ,

○○○○○ v1 v2 v3 ○○○○

The D3DTRIFLAG_ODD D3DTRIFLAG_EVEN
○○○○ 5 ○○○

D3DTRIFLAG_START
D3DTRIFLAG_ODD
D3DTRIFLAG_EVEN
D3DTRIFLAG_ODD
D3DTRIFLAG_EVEN

○ 5 ○

D3DTRIFLAG_START
D3DTRIFLAG_EVEN
D3DTRIFLAG_EVEN
D3DTRIFLAG_EVEN

D3DTRIFLAG_EVEN

5 の の の

D3DTRIFLAG_STARTFLAT(4)

D3DTRIFLAG_EVEN

D3DTRIFLAG_EVEN

D3DTRIFLAG_EVEN

D3DTRIFLAG_EVEN

の

D3DVECTOR

```
typedef struct _D3DVECTOR {
    union {
        D3DVALUE x;
        D3DVALUE dvX;
    };
    union {
        D3DVALUE y;
        D3DVALUE dvY;
    };
    union {
        D3DVALUE z;
        D3DVALUE dvZ;
    };
} D3DVECTOR, *LPD3DVECTOR;
```

の Direct3D Direct3DRM の

dvX dvY dvZ

ffff,Ž',,D3DVALUE€,'

***D3DLIGHT D3DLIGHTINGELEMENT D3DRMBOX
D3DRMQUATERNION D3DRMVERTEX***

D3DVERTEX

```
typedef struct _D3DVERTEX {
    union {
        D3DVALUE x;
        D3DVALUE dvX;
    };
    union {
        D3DVALUE y;
        D3DVALUE dvY;
    };
    union {
        D3DVALUE z;
        D3DVALUE dvZ;
    };
    union {
```

```

        D3DVALUE nx;
        D3DVALUE dvNX;
    };
    union {
        D3DVALUE ny;
        D3DVALUE dvNY;
    };
    union {
        D3DVALUE nz;
        D3DVALUE dvNZ;
    };
    union {
        D3DVALUE tu;
        D3DVALUE dvTU;
    };
    union {
        D3DVALUE tv;
        D3DVALUE dvTV;
    };
};
} D3DVERTEX, *LPD3DVERTEX;

```

(D)

D3DOPCODE の D3DOP_TRIANGLE

```

dvX dvY dvZ
    <Ž, •,“,Ž’,,D3DVALUEE,’
dvNX dvNY dvNZ
    < •,“,Ž’,,D3DVALUE E,’
dvTU □ dvTV
    ’“,fffff,Ž’,,D3DVALUEE,’
    D3DVALUE

```

D3DVIEWPORT

```

typedef struct _D3DVIEWPORT {
    DWORD    dwSize;
    DWORD    dwX;
    DWORD    dwY;
    DWORD    dwWidth;
    DWORD    dwHeight;
    D3DVALUE dvScaleX;
    D3DVALUE dvScaleY;
    D3DVALUE dvMaxX;
    D3DVALUE dvMaxY;
    D3DVALUE dvMinZ;
    D3DVALUE dvMaxZ;
} D3DVIEWPORT, *LPD3DVIEWPORT;

```

3D 2D 3D
IDirect3DViewport::GetViewport **IDirect3DViewport::SetViewport**

dwSize

,, “,fff”^,fff

dwX □ **dwY**

ff f f, <, •

dwWidth □ **dwHeight**

ff f f,”—

dvScaleX □ **dvScaleY**

fff f,<Ž,ff f”Ž’,,D3DVALUE€,’

dvMaxX **dvMaxY** **dvMinZ** **dvMaxZ**

x y z,<Ž •, “, ’,Ž’,,D3DVALUE€,’

D3DVALUE *IDirect3DViewport::GetViewport*
IDirect3DViewport::SetViewport

**D3DBLEND**

```
typedef enum _D3DBLEND {
    D3DBLEND_ZERO           = 1,
    D3DBLEND_ONE           = 2,
    D3DBLEND_SRCCOLOR      = 3,
    D3DBLEND_INVSRCCOLOR   = 4,
    D3DBLEND_SRCALPHA      = 5,
    D3DBLEND_INVSRCALPHA   = 6,
    D3DBLEND_DESTALPHA     = 7,
    D3DBLEND_INVDESTALPHA  = 8,
    D3DBLEND_DESTCOLOR     = 9,
    D3DBLEND_INVDESTCOLOR  = 10,
    D3DBLEND_SRCALPHASAT   = 11,
    D3DBLEND_BOTHSRCALPHA  = 12,
    D3DBLEND_BOTHINVSRCALPHA = 13,
} D3DBLEND;
```

D3DRENDERSTATETYPE のの **D3DRENDERSTATE_DESTBLEND** のの
 のの RGBA s□d

D3DBLEND_ZERO

—‘, (0, 0, 0, 0)

D3DBLEND_ONE

—‘, (1, 1, 1, 1)

D3DBLEND_SRCCOLOR

—‘, (Rs, Gs, Bs, As)

D3DBLEND_INVSRCOLOR

—‘, (As, As, As, As, 1-As)

D3DBLEND_SRCALPHA

—‘, (As, As, As, As)

D3DBLEND_INVSRALPHA

—‘, (1-As, 1-As, 1-As)

D3DBLEND_DESTALPHA

—‘, (Ad, Ad, Ad, Ad)

D3DBLEND_INVDESTALPHA

—‘, (1-Ad, 1-Ad, 1-Ad, 1-Ad)

D3DBLEND_DESTCOLOR

—‘, (Rd, Gd, Bd, Ad)

D3DBLEND_INVDESTCOLOR

—‘, (1-Rd, 1-Gd, 1-Bd, 1-Ad)

D3DBLEND_SRCALPHASAT

—‘, (f, f, f, 1) f = min(As, 1-Ad)

D3DBLEND_BOTHSRCALPHA*f f* —‘, (As, As, As, As) *fffff fff* —‘, (1-As, 1-As, 1-As, 1-As) *fffff fff*, -E,,**D3DBLEND_BOTHINVSRCALPHA***f f* —‘, (1-As, 1-As, 1-As, 1-As) *fffff fff* —‘, (As, As, As, As) *fffff fff*, -E,,**D3DCMPFUNC**

```
typedef enum _D3DCMPFUNC {
    D3DCMP_NEVER           = 1,
    D3DCMP_LESS            = 2,
    D3DCMP_EQUAL           = 3,
    D3DCMP_LESSEQUAL      = 4,
    D3DCMP_GREATER        = 5,
    D3DCMP_NOTEQUAL       = 6,
    D3DCMP_GREATEREQUAL   = 7,
    D3DCMP_ALWAYS         = 8,
} D3DCMPFUNC;
```

D3DRENDERSTATETYPE の D3DRENDERSTATE_ZFUNC**D3DRENDERSTATE_ALPHAFUNC の****D3DCMP_NEVER**,,*fff*,Z”,,**D3DCMP_LESS**,,*ffff*’, E ,*ffff*’,, ,,,,%,,**D3DCMP_EQUAL**,,*ffff*’, E ,*ffff*’,“,,,,%,,

```

D3DCMP_LESSEQUAL
    ,ffff', @ ,ffff'^%o,,,%o,,
D3DCMP_GREATER
    ,ffff', @ ,ffff',',,,,,%o,,
D3DCMP_NOTEQUAL
    ,ffff', @ ,ffff',",,,,,,%o,,
D3DCMP_GREATEREQUAL
    ,ffff', @ ,ffff'^ ,,,,%o,,
D3DCMP_ALWAYS
    ,,,fff,Ž ,,

```

D3DCOLORMODEL

```

typedef enum _D3DCOLORMODEL {
    D3DCOLOR_MONO = 1,
    D3DCOLOR_RGB = 2,
} D3DCOLORMODEL;

```

のの

```

D3DCOLOR_MONO
    fffffff(,, fffffff),Ž—,, ,fff, "“ , ,', "“,<,"<,,,,Ž—,,
D3DCOLOR_RGB
    Š',RGBffff,Ž—,,
    D3DDEVICEDESC D3DFINDDEVICESEARCH
D3DLIGHTSTATETYPE IDirect3DRMDevice::GetColorModel

```

D3DCULL

```

typedef enum _D3DCULL {
    D3DCULL_NONE = 1,
    D3DCULL_CW = 2,
    D3DCULL_CCW = 3,
} D3DCULL;

```

のの

```

D3DCULL_NONE
    ,,,,,
D3DCULL_CW
    %o%o,, ,",
D3DCULL_CCW
    %o,, ,",
    D3DPRIMCAPS D3DRENDERSTATETYPE

```

D3DFILLMODE

```

typedef enum _D3DFILLMODE {
    D3DFILL_POINT = 1,

```

```

    D3DFILL_WIREFRAME = 2,
    D3DFILL_SOLID      = 3
} D3DFILLMODE;

```

のD3DRENDERSTATETYPE のD3DRENDERSTATE_FILLMODE の

D3DFILL_POINT

“ “ ” ”

D3DFILL_WIREFRAME

fff ff f, “ ” ” ”

D3DFILL_SOLID

— “ ” ” ”

D3DFOGMODE

```

typedef enum _D3DFOGMODE {
    D3DFOG_NONE      = 0,
    D3DFOG_EXP       = 1,
    D3DFOG_EXP2      = 2,
    D3DFOG_LINEAR    = 3
} D3DFOGMODE;

```

D3DRENDERSTATETYPE のD3DRENDERSTATE_FOGTABLEMODE の

D3DFOG_NONE

fffE%, “ ” ” ”

D3DFOG_EXP

Ž, •Ž, ‘ , fffE%, ‘,

$$f = e^{-(density \times z)}$$

D3DFOG_EXP2

Ž, •Ž, ‘ 2 , fffE%, ‘,

$$f = e^{-(density \times z)^2}$$

D3DFOG_LINEAR

Ž, •Ž, Ž“ , “Š, E, fffE%, ‘,

$$f = \frac{end - z}{end - start}$$

„, E ff f, „„, —, fff f f, „„

D3DLIGHTSTATETYPE

```
typedef enum _D3DLIGHTSTATETYPE {
    D3DLIGHTSTATE_MATERIAL = 1,
    D3DLIGHTSTATE_AMBIENT = 2,
    D3DLIGHTSTATE_COLORMODEL = 3,
    D3DLIGHTSTATE_FOGMODE = 4,
    D3DLIGHTSTATE_FOGSTART = 5,
    D3DLIGHTSTATE_FOGEND = 6,
    D3DLIGHTSTATE_FOGDENSITY = 7,
} D3DLIGHTSTATETYPE;
```

D3DOP_STATELIGHT の D3DSTATE の

D3DLIGHTSTATE_MATERIAL

Ⓔ, ,fff%,Š,‘,“ ,Ž ,,,Ž—,, —,“,,,ffffff,‘<, fffff’,NULL,,,

D3DLIGHTSTATE_AMBIENT

,Ⓔ ,ⒺⒺ,“ , , ‘,, ffff fff,,,‘Ž’,,,,, fff f,,ⒺⒺ,Ž’,,, fffff’,0
,,

D3DLIGHTSTATE_COLORMODEL

D3DCOLORMODEL—Ⓔ,fff,,, fffff’,D3DCOLOR_RGB,,,

D3DLIGHTSTATE_FOGMODE

D3DFOGMODE—Ⓔ,fff,,, fffff’,D3DFOG_NONE,,,

D3DLIGHTSTATE_FOGSTART

fff,ŠŽ’,<, fffff’,1.0,,,

D3DLIGHTSTATE_FOGEND

fff, —’<, fffff’,100.0,,,

D3DLIGHTSTATE_FOGDENSITY

fff, ‘, “,<, fffff’,1.0,,,

D3DOPCODE □ *D3DSTATE*

D3DLIGHTTYPE

```
typedef enum _D3DLIGHTTYPE {
    D3DLIGHT_POINT = 1,
    D3DLIGHT_SPOT = 2,
    D3DLIGHT_DIRECTIONAL = 3,
    D3DLIGHT_PARALLELPOINT = 4,
    D3DLIGHT_GLSPOT = 5,
} D3DLIGHTTYPE;
```

の D3DLIGHT の

D3DLIGHT_POINT

ⒺⒺ, “,,,

D3DLIGHT_SPOT

ⒺⒺ,ffffff,,,

D3DLIGHT_DIRECTIONAL

ⒺⒺ, ‘ Ⓔ,,,

D3DOP_STATERENDER

```

fffffff f—
,ffffff ffff,“• ‘• ,’, ’,, fffff f f, • ,,,f ff, ,,’,,, f ff, “
• ‘• ,Ž, ,,• ’, ’,, ’,, ’,, ,,,• ,,,, , D3DSTATE “,D3DREN
DERSTATETYPE—(E,Ž ,,,

```

D3DOP_PROCESSVERTICES

```

“,(E,Š, ’,, fffff f f, D3DPROCESSVERTICES “,Ž’,,,

```

D3DOP_TEXTURELOAD

```

fffffff ffff,f f,“,,fff ,,, fffff f f, D3DTEXTURELOAD “,
Ž’,,,

```

D3DOP_EXIT

```

“,,fff, —,“,,Ž,Ž,

```

D3DOP_BRANCHFORWARD

```

Ž ffff,•Š,%o”,,, , D3DBRANCH “,Ž ,,,,

```

D3DOP_SPAN

```

“,Y’“,fff,fff’, , D3DSPAN “,Ž ,,,,

```

D3DOP_SETSTATUS

```

Ž ffff, ‘,ffff,, , D3DSTATUS “,Ž ,,,,

```

```

O( ) /OOOOO

```

D3DINSTRUCTION**D3DRENDERSTATETYPE**

```

typedef enum _D3DRENDERSTATETYPE {
    D3DRENDERSTATE_TEXTUREHANDLE = 1,
    D3DRENDERSTATE_ANTIALIAS = 2,
    D3DRENDERSTATE_TEXTUREADDRESS = 3,
    D3DRENDERSTATE_TEXTUREPERSPECTIVE = 4,
    D3DRENDERSTATE_WRAPU = 5,
    D3DRENDERSTATE_WRAPV = 6,
    D3DRENDERSTATE_ZENABLE = 7,
    D3DRENDERSTATE_FILLMODE = 8,
    D3DRENDERSTATE_SHADEMODE = 9,
    D3DRENDERSTATE_LINEPATTERN = 10,
    D3DRENDERSTATE_MONOENABLE = 11,
    D3DRENDERSTATE_ROP2 = 12,
    D3DRENDERSTATE_PLANEMASK = 13,
    D3DRENDERSTATE_ZWRITEENABLE = 14,
    D3DRENDERSTATE_ALPHATESTENABLE = 15,
    D3DRENDERSTATE_LASTPIXEL = 16,
    D3DRENDERSTATE_TEXTUREMAG = 17,
    D3DRENDERSTATE_TEXTUREMIN = 18,
    D3DRENDERSTATE_SRCBLEND = 19,
    D3DRENDERSTATE_DESTBLEND = 20,
    D3DRENDERSTATE_TEXTUREMAPBLEND = 21,
    D3DRENDERSTATE_CULLMODE = 22,
    D3DRENDERSTATE_ZFUNC = 23,
    D3DRENDERSTATE_ALPHAREF = 24,
    D3DRENDERSTATE_ALPHAFUNC = 25,

```

```

D3DRENDERSTATE_DITHERENABLE = 26,
D3DRENDERSTATE_BLENDENABLE = 27,
D3DRENDERSTATE_FOGENABLE = 28,
D3DRENDERSTATE_SPECULARENABLE = 29,
D3DRENDERSTATE_ZVISIBLE = 30,
D3DRENDERSTATE_SUBPIXEL = 31,
D3DRENDERSTATE_SUBPIXELX = 32,
D3DRENDERSTATE_STIPPLEDALPHA = 33,
D3DRENDERSTATE_FOGCOLOR = 34,
D3DRENDERSTATE_FOGTABLEMODE = 35,
D3DRENDERSTATE_FOGTABLESTART = 36,
D3DRENDERSTATE_FOGTABLEEND = 37,
D3DRENDERSTATE_FOGTABLEDENSITY = 38,
D3DRENDERSTATE_STIPPLEENABLE = 39,
D3DRENDERSTATE_STIPPLEPATTERN00 = 64,
// "\*ff□f,01,,30,□-
D3DRENDERSTATE_STIPPLEPATTERN31 = 95,
} D3DRENDERSTATETYPE;

```

D3DOP_STATERENDER のD3DSTATE のののの2の

D3DRENDERSTATE_TEXTUREHANDLE

```
ffff,ffff ffff',NULL,,
```

D3DRENDERSTATE_ANTIALIAS

```
fffffffff—,' ffff',FALSE,,
```

D3DRENDERSTATE_TEXTUREADDRESS

D3DTEXTUREADDRESS

```
—(E,fff,, ffff',D3DTEXTUREADDRESS_WRAP,,
```

D3DRENDERSTATE_TEXTUREPERSPECTIVE

```
TRUE,, “Ž ffff',FALSE,,
```

D3DRENDERSTATE_WRAPU

```
TRUE,, u(E,'*, ffff',FALSE,,
```

D3DRENDERSTATE_WRAPV

```
TRUE,, v(E,'*, ffff',FALSE,,
```

D3DRENDERSTATE_ZENABLE

```
TRUE,, Zffff,"Šfff,%", ffff',FALSE,,
```

D3DRENDERSTATE_FILLMODE

```
D3DFILLMODE—(E,fff,, ffff',D3DFILL_SOLID,,
```

D3DRENDERSTATE_SHADEMODE

```
D3DSHADEMODE—(E,fff,, ffff',D3DSHADE_GOURAUD,,
```

D3DRENDERSTATE_LINEPATTERN

```
D3DLINEPATTERN “,, ffff',wRepeatPattern—,0, wLinePattern—,0,,
```

D3DRENDERSTATE_MONOENABLE

```
TRUE,, ffff fffffff,%", ffff',FALSE,, ffff,RGBffffff,ff
f,,,,, TRUE,, ffff fff, ffff,RGBffffff,ff f,,,,,D3DDEVICE
DESC “,dcmColorModelfff,Ž—,',,,,,
```


D3DRENDERSTATE_DITHERENABLE

TRUE,,,fffff,%o",,, fffff',FALSE,,,

D3DRENDERSTATE_BLENDENABLE

TRUE,,,ffff, %o",,, fffff',FALSE,,,

D3DRENDERSTATE_FOGENABLE

TRUE,,,fff,%o",,, fffff',FALSE,,,

D3DRENDERSTATE_SPECULARENABLE

TRUE,,,“%o,%o”,,, fffff',TRUE,,,

D3DRENDERSTATE_ZVISIBLE

TRUE,,,ZC %o",,, fffff',FALSE,, ZC , , *%o
,,ffff,C,,,,Zffff,',,fff,, ,,,ffff,ff f,fff f<Š,*,,ffff,, ,’<,,,,,,

D3DRENDERSTATE_SUBPIXEL

TRUE,,,fffff,' %o",,, fffff',FALSE,,,

D3DRENDERSTATE_SUBPIXELX

TRUE,,,X,,,' %o",,, fffff',FALSE,,,

D3DRENDERSTATE_STIPPLEDALPHA

TRUE,,,“*,ffff,%o”,,, fffff',FALSE,,,

D3DRENDERSTATE_FOGCOLOR

D3DCOLORC,' fffff',0,,,

D3DRENDERSTATE_FOGTABLEMODE

D3DFOGMODE—(C,fff,, fffff',D3DFOG_NONE,,,

D3DRENDERSTATE_FOGTABLESTART

fff f ff,ŠŽ ,,, Cfff f f,,ŠŽ,,fff(C%o,^,,,

D3DRENDERSTATE_FOGTABLEEND

fff f ff, — ,,, Cfff f f, ‘-“,“,,,fff(C%o,^,,,

D3DRENDERSTATE_FOGTABLEDENSITY

Cfff f f, ‘fff-“, ', , ,’^, 0,,1,,,

D3DRENDERSTATE_STIPPLEENABLE

ffff ffff,“*,%o”,,, “*,ffff,%o”,,, C ,“ff f,-(C,,,,,,),,,

D3DRENDERSTATE_STIPPLEPATTERN00

D3DRENDERSTATE_STIPPLEPATTERN31

“*ff f ,,,,ffffff ‘, “*ff f,*— ,“%o,,,

D3DOPCODE D3DSTATE

D3DSHADEMODE

```
typedef enum _D3DSHADEMODE {  
    D3DSHADE_FLAT = 1,  
    D3DSHADE_GOURAUD = 2,  
    D3DSHADE_PHONG = 3,  
} D3DSHADEMODE;
```

D3DRENDERSTATETYPE 的 D3DRENDERSTATE_SHADEMODE

D3DSHADE_FLAT

```
•-ff ffff f f ŽŠŒ, ,“ , •-, ,Œ’,,,,Ž—,„
```

D3DSHADE_GOURAUD

```
ff ff ffff f f •-, , ŽŠŒ,,,“Š, , ,, Œ,,,Œ’,,,
```

D3DSHADE_PHONG

```
fff ff ffff f f ,f f, Œ ff f,,,,,
```

D3DRENDERSTATETYPE

D3DTEXTUREADDRESS

```
typedef enum _D3DTEXTUREADDRESS {
    D3DTEXTUREADDRESS_WRAP = 1,
    D3DTEXTUREADDRESS_MIRROR = 2,
    D3DTEXTUREADDRESS_CLAMP = 3,
} D3DTEXTUREADDRESS;
```

D3DRENDERSTATETYPE の**D3DRENDERSTATE_TEXTUREADDRESS の****D3DTEXTUREADDRESS_WRAP**

D3DRENDERSTATETYPE—ŒŒ,

D3DRENDERSTATE_WRAPU,**D3DRENDERSTATE_WRAPV**fffff
,Ž—,„ ,, ffff, ’,,

D3DTEXTUREADDRESS_MIRROR

ffff ffff,fff,“, (D3DRENDERSTATE_WRAPU,,D3DRENDERSTATE_WRAPV,,,,)

,,, ,, fffff,,ffff, , ,,ffff, 0,1,Š,u’, <fff,, 1,2, Š,, ffff(“),,, ,,2,3,Š,, , <-,

D3DTEXTUREADDRESS_CLAMP

ffff •,1.0,‘,,, 1.0, ’,, 0.0, ,, 0.0, ’,,

D3DRENDERSTATE_WRAPU **D3DRENDERSTATE_WRAPV**

IDirect3DTexture

D3DRENDERSTATETYPE

D3DTEXTUREBLEND

```
typedef enum _D3DTEXTUREBLEND {
    D3DTEXTUREBLEND_DECAL = 1,
    D3DTEXTUREBLEND_MODULATE = 2,
    D3DTEXTUREBLEND_DECALALPHA = 3,
    D3DTEXTUREBLEND_MODULATEALPHA = 4,
    D3DTEXTUREBLEND_DECALMASK = 5,
    D3DTEXTUREBLEND_MODULATEMASK = 6,
    D3DTEXTUREBLEND_COPY = 7,
} D3DTEXTUREBLEND;
```

のD3DRENDERSTATETYPE の**D3DRENDERSTATE_TEXTUREMAPBLEND**

D3DTBLEND_DECAL

$ff f fffff f f,ff f,, ,,f f,, fffff,RGB,ffff', fffff,Ž—$
 $,,,,, ,Š,,$

D3DTBLEND_MODULATE

$' fffff f f,ff f,, ,,f f,, fffff,RGB', fffff,Ž—$
 $,,,,,RGB',E,•, fffff,ffff', fffff,Ž—,,,,, ,ffff',Š,,$

D3DTBLEND_DECALALPHA

$ff f ffff fffff f f,ff f,, ,,f f,, fffff,RGB,ffff', Ž,•Ž,$
 $fffff,Ž—,,,,, , ,,,$

$$C = (1 - A_t) C_o + A_t C_t$$

$,,•Ž, C, A,ffff t,fffff o,E,ffffff('),•,$

$D3DTBLEND_DECALALPHAf f, fffff,ffff', fffff,Ž—$
 $,,,,, ,ffff',Š,,$

D3DTBLEND_MODULATEALPHA

$•ffff,fffff f f,ff f,, ,,f f, fffff,RGB', RGB',',, fffff,ff$
 $ff',',,$

D3DTBLEND_DECALMASK

$ff f fff,fffff f f,ff f,,,$

D3DTBLEND_MODULATEMASK

$•fff,fffff f f,ff f,,,$

D3DTBLEND_COPY

$•Žfffff f f,ff f,,,$

$00 10000000(1)$
 $(1 1 1)00$

D3DTEXTUREFILTER

```
typedef enum _D3DTEXTUREFILTER {
    D3DFILTER_NEAREST           = 1,
    D3DFILTER_LINEAR            = 2,
    D3DFILTER_MIPNEAREST       = 3,
    D3DFILTER_MIPLINEAR        = 4,
    D3DFILTER_LINEAR_MIPNEAREST = 5,
    D3DFILTER_LINEAR_MIPLINEAR = 6,
} D3DTEXTUREFILTER;
```

D3DRENDERSTATETYPE の D3DRENDERSTATE_TEXTUREMAG

D3DFILTER_NEAREST

$—,.,ffff', ,<, •ffff,Ž—,, ,,, Š', ,—•,—$
 $,,,, Š', ,ff f,,,,,, —•,ff f,,,,,,$

D3DFILTER_LINEAR

$2 2,—, ,•< ffffŠ<—,.,ffff,Ž—,, ,,,Š', ,—•,—$
 $,,,, Š', ,ff f,,,,,, —•,ff f,,,,,,$

D3DERR_BADMINORVERSION
D3DERR_EXECUTE_CLIPPED_FAILED
D3DERR_EXECUTE_CREATE_FAILED
D3DERR_EXECUTE_DESTROY_FAILED
D3DERR_EXECUTE_FAILED
D3DERR_EXECUTE_LOCK_FAILED
D3DERR_EXECUTE_LOCKED
D3DERR_EXECUTE_NOT_LOCKED
D3DERR_EXECUTE_UNLOCK_FAILED
D3DERR_LIGHT_SET_FAILED
D3DERR_MATERIAL_CREATE_FAILED
D3DERR_MATERIAL_DESTROY_FAILED
D3DERR_MATERIAL_GETDATA_FAILED
D3DERR_MATERIAL_SETDATA_FAILED
D3DERR_MATRIX_CREATE_FAILED
D3DERR_MATRIX_DESTROY_FAILED
D3DERR_MATRIX_GETDATA_FAILED
D3DERR_MATRIX_SETDATA_FAILED
D3DERR_SCENE_BEGIN_FAILED
D3DERR_SCENE_END_FAILED
D3DERR_SCENE_IN_SCENE
D3DERR_SCENE_NOT_IN_SCENE
D3DERR_SETVIEWPORTDATA_FAILED
D3DERR_TEXTURE_CREATE_FAILED
D3DERR_TEXTURE_DESTROY_FAILED
D3DERR_TEXTURE_GETSURF_FAILED
D3DERR_TEXTURE_LOAD_FAILED
D3DERR_TEXTURE_LOCK_FAILED
D3DERR_TEXTURE_LOCKED
D3DERR_TEXTURE_NO_SUPPORT

D3DERR_TEXTURE_NOT_LOCKED

D3DERR_TEXTURE_SWAP_FAILED

D3DERR_TEXTURE_UNLOCK_FAILED