

Part C

Microsoft® DirectX™ 3

ffffffŠ"fff

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

©1996 Microsoft Corporation. All rights reserved.

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

„, „, •, %Ž-„, ŠŽ, •„„

Direct3D' f f f f f f f f

f f f
 f f f f Š
 IDirect3D.....
 IDirect3DDevice.....
 IDirect3DExecuteBuffer.....
 IDirect3DLight.....
 IDirect3DMaterial.....
 IDirect3DTexture.....
 IDirect3DViewport.....
 “
 —Œ
 , , Œ
 •'

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),,,fff(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)

- ŠĖ,’,•,

val
•Š,,,’,

prec
–Ė

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*
- ,Ž, , ,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 ,•Š,,,RGBA
- RGB_TORGBA*

f ffffŠ

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
`—<,Œ',,`

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)
(LPDDSDSURFACEDESC 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,fffff

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,,, IDirect3DLightfff ffff,,ffff, ’,,ffff

pUnkOuter

```
„fff f,COM %o“, —
ŒŠ ,Ž,,,,,,, IDirect3D::CreateLightffff,,fff f,NULL^Š,,,ff ,•,
```

IDirect3D::CreateMaterial

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

Direct3DMaterial

- Œ,,, D3D_OK,•, •,^—, Direct3D ' f f,•' ,Ž ,,,,

lpDirect3DMaterial

```
Œ, ,, Œ,,, 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 の IDirect3DDevice::AddViewport

Direct3D

- Œ,,,D3D_OK Ž”,,, Ž',,,,,,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

lpD3DViewport

```
Œ, ,, Œ,,, 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

- Œ,,,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(LPD3DFINDDEVICESEARCH lpD3DFDS,
LPD3DFINDDEVICERESULT 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‰
,,,, 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	の COM
<i>IUnknown</i>	IUnknown の
AddRef	
QueryInterface	
Release	
Direct3DDevice	3D
IDirect3DDevice::QueryInterface	DirectDrawSurface

IDirect3DDevice::AddViewport

HRESULT AddViewport (LPDIRECT3DVIEWPORT lpDirect3DViewport);

の

- D3D_OK, DDERR_INVALIDOBJECT, DDERR_INVALIDPARAMS

lpDirect3DViewport

Direct3DDevice, IDirect3DViewport, ffff, ffff, ffff

IDirect3DDevice::BeginScene

HRESULT BeginScene();

- D3D_OK,

IDirect3DDevice::EndScene

```
HRESULT CreateExecuteBuffer(LPDIRECT3DEXECUTEBUFFERDESC lpDesc,
    LPDIRECT3DEXECUTEBUFFER* lplpDirect3DExecuteBuffer,
    IUnknown* pUnkOuter);
```

- $\mathbb{E}_{,,,D3D_OK} \check{Z}''_{,,,} \check{Z}'_{,,,,,,,},$

DDERR_INVALIDPARAMS

```

,,,Direct3DExecuteBufferfffff,Ž,D3DEXECUTEBUFFERDESC “
,,ffff ,,ffff,fff,‘,,ffffff, ,,,,,, ,E, ,,Ž”,,

```

```

,,Direct3DExecuteBufferfffff,,ffff,,ffff

```

```

,,fff f, COM“, —
Š,,,,,,,, IDirect3DDevice::CreateExecuteBufferffff, ,,fff f,NULL
^Š,,ff ,.

```

IDirect3DDevice9::Lock の

```
HRESULT CreateMatrix(LPD3DMATRIXHANDLE lpD3DMatHandle);
```

DDERR_INVALIDPARAMS

,,, —,ffff• ,ffff ffff,fff,‘,,, —, ,,,,,, ,ℰ, „Ž”,

```
HRESULT DeleteMatrix(D3DMATRIXHANDLE d3dMatHandle);
```

IDirect3DDevice::CreateMatrix

- D3D_OK, D3D_ERR_INVALIDPARAMS

d3dMatHandle

,,, ,ffff

IDirect3DDevice::CreateMatrix IDirect3DDevice::SetMatrix

IDirect3DDevice::DeleteViewport

HRESULT DeleteViewport(LPDIRECT3DVIEWPORT lpDirect3DViewport);

○

- D3D_OK, D3D_ERR_INVALIDOBJECT, D3D_ERR_INVALIDPARAMS

lpDirect3DViewport

Direct3DDevicefffff,, ,Direct3DViewportfffff,ffff

IDirect3DDevice::EndScene

HRESULT EndScene();

IDirect3DDevice::BeginScene ○

- D3D_OK, D3D_ERR_INVALIDPARAMS

IDirect3DDevice::BeginScene

IDirect3DDevice::EnumTextureFormats

HRESULT EnumTextureFormats(
LPD3DENUMTEXTUREFORMATSCALLBACK lpd3dEnumTextureProc,
LPVOID lpArg);

○○

- D3D_OK, D3D_ERR_INVALIDOBJECT, D3D_ERR_INVALIDPARAMS

lpd3dEnumTextureProc

—< —

,,,ŠfffffD3DENUMTEXTUREFORMATSCALLBACKf ff
ffŠ ,ffff

lpArg
f ffffŠ ,“,,,ffff fff’<f f,,ffff

IDirect3DDevice::Execute

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

- Ą,,,D3D_OK Ź”,,, Ź,’,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpDirect3DExecuteBuffer
Ź ,,,Ź ffff,,ffff
lpDirect3DViewport
•Ź,,,Ź ffff,•Šffffff,< ,Direct3DViewportffffff,,ffff
dwFlags
ffff,ffffff,ffffff,,,Ź’,,fff ,fff f, Ź,’,,,,,,
D3DEXECUTE_CLIPPED
ff f f,Š,,,ffffff,ffffff,,
D3DEXECUTE_UNCLIPPED
ff f f,’,,,,ffffff,Š,(ffffff,,)

D3DEXECUTEDATA D3DINSTRUCTION
IDirect3DExecuteBuffer::Validate

IDirect3DDevice::GetCaps

HRESULT GetCaps(LPDDDEVICEDESC lpD3DHWDevDesc,
LPDDDEVICEDESC lpD3DHELDevDesc);

Direct3DDevice の

- Ą,,,D3D_OK Ź”,,, Ź,’,,,,,•,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpD3DHWDevDesc
ffff,f ffff •,Š,D3DDEVICEDESC “,,ffff
lpD3DHELDevDesc
—,,,,,ffffff,ffff fff •,Š,D3DDEVICEDESC “,,ffff
のののIDirectDraw2::GetCaps

IDirect3DDevice::GetDirect3D

```
HRESULT GetDirect3D(LPDIRECT3D* lpD3D);
```

④ IDirect3D

- $\mathbb{E}_{D3D_OK}, \text{ , , , , , } f f \text{ , , , } \bullet \text{ , , , , } Direct3D \text{ ' } f \text{ } f \text{ , } \bullet \text{ , } \checkmark \text{ , , , ,}$

 $lpD3D$

ffff,•,fff ffff,,ffff

IDirect3DDevice::GetMatrix

```
HRESULT GetMatrix(D3DMATRIXHANDLE lpD3DMatHandle,  
LPD3DMATRIX lpD3DMatrix);
```

④IDirect3DDevice::CreateMatrix

 \mathcal{O}

- $\mathbb{E}_{,,,D3D_OK} \check{Z}''_{,,,} \check{Z}_{,}',.$

DDERR_INVALIDPARAMS

lpD3DMatHandle

Ž“., —•.,ffff

lpD3DMatrix

ffff,•, —,Š,D3DMATRIX “.,ffff

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

IDirect3DDevice::SetMatrix

IDirect3DDevice::GetPickRecords

```
HRESULT GetPickRecords(LPDWORD lpCount,
    LPD3DPICKRECORD lpD3DPickRec);
```

- $\mathbb{E}_{,,D3D_OK},$

lpCount

Ž“ „D3DPICKRECORD “, ,“,• „ffff

 $lpD3DPickRec$

ffff,•,D3DPICKRECORD “,”—,,ffff



IDirect3DDevice::GetStats

```
HRESULT GetStats(LPD3DSTATS lpD3DStats);
```

 \mathcal{O}

- $\mathbb{E}_{,,,D3D_OK} \check{Z}_{,,,} \check{Z}_{',,,,,,},$

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

lpD3DStats
•,Ž,D3DSTATS “,,ffff

IDirect3DDevice::Initialize

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

- Ą,,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* lplpDirect3DViewport, DWORD dwFlags);
```

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

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

lplpDirect3DViewport
Direct3DDeviceffffff,Š~,ff f f,fff,’Ž,ff f f,,ffff

dwFlags
ff f f,fff,,Ž“,,ff f f,Ž’,,fff fffff,D3DNEXT_NEXT
D3DNEXT_HEAD fff, , –,Ž“
D3DNEXT_NEXT fff,Ž, –,Ž“
D3DNEXT_TAIL fff, Ą, –,Ž“

IDirect3DDevice::Pick

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

Direct3DExecuteBuffer ⑦

- lpDirect3DExecuteBuffer*

lpDirect3DViewport

dwFlags

 $lpRect$ $lpRect \mathcal{O} \mathbf{x1} \quad \mathbf{x2}$

y1 y2

Direct3DExecuteBuffer の Direct3DDevice

IDirect3DDevice::SetMatrix

④IDirect3DDevice::CreateMatrix

- d3dMatHandle*

lpD3DMatrix

④ IDirect3DDevice::SetMatrix

□ □ □ □ □

 \mathcal{O} **IDirect3DDevice::CreateMatrix** **IDirect3DDevice::GetMatrix**

IDirect3DDevice::DeleteMatrix

IDirect3DDevice::SwapTextureHandles

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

の

- D3D_OK,*,

lpD3DTex1 lpD3DTex2
ffff,-,,, D3D_OK,ffff,ffff,,ffff

のののの

IDirect3DExecuteBuffer

IDirect3DExecuteBuffer のDirect3D ののののIDirect3DDeviceExecuteBuffer

IDirect3DExecuteBuffer のの

f f,Ž

GetExecuteData

SetExecuteData

fff,%o

Lock

Unlock

,,

Initialize

Optimize

Validate

IDirect3DExecuteBuffer

の COM

IUnknown

IUnknown の

AddRef

QueryInterface

Release

IDirect3DExecuteBuffer::GetExecuteData

HRESULT GetExecuteData(LPD3DEXECUTEDATA lpData);

Direct3DExecuteBuffer ののDirect3DExecuteBuffer の

- D3D_OK Ž”,,, Ž,’,,,,*,
D3DERR_EXECUTE_LOCKED
DERR_INVALIDOBJECT
DERR_INVALIDPARAMS

lpData

Direct3DExecuteBufferfffff,D3D_OK Ž f f, ‘,’,,D3DEXECUTEDAT
A “,,ffff

Direct3DExecuteBuffer の

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

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

COM の

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

lpDirect3DDevice

Direct3D fffffff,Ž,ffff,,ffff

lpDesc

```
,,,Direct3DExecuteBufferfffff,‹ ,D3DEXECUTEBUFFERDESC
“,,ffff ffff,fff,‘,,,,, Œ, ,,Ž”,,
```

IDirect3DExecuteBuffer::Lock

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

のの

- Œ,,,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 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 IDirect3DLight の IDirect3DLight

IDirect3DLight の

GetLight
SetLight

Initialize

IDirect3DLight*IUnknown*

の COM

の

IUnknown の**AddRef****QueryInterface****Release****IDirect3DLight::GetLight**`HRESULT GetLight(LPD3DLIGHT lpLight);`

Direct3DLight の

- `S_OK`, `D3D_OK` 成功, `S_FALSE`, `D3DERR_INVALIDOBJECT`, `D3DERR_INVALIDPARAMS`

lpLight`LPD3DLIGHT`*IDirect3DLight::SetLight***IDirect3DLight::Initialize**`HRESULT Initialize(LPDIRECT3D lpDirect3D);`

COM の の の

- `S_OK`, `D3D_OK` 成功, `S_FALSE`, `D3DERR_ALREADY_INITIALIZED`, `D3DERR_INVALIDOBJECT`, `D3DERR_INVALIDPARAMS`

lpDirect3D`LPDIRECT3D`**IDirect3DLight::SetLight**`HRESULT SetLight(LPD3DLIGHT lpLight);`

Direct3DLight の

- `S_OK`, `D3D_OK` 成功, `S_FALSE`, `D3DERR_INVALIDOBJECT`, `D3DERR_INVALIDPARAMS`

*lpLight*の `D3DLIGHT` の*IDirect3DLight::GetLight*

IDirect3DMaterial

IDirect3DMaterial の IDirect3Dmaterial
IDirect3DMaterial

IDirect3DMaterial の

, — **Reserve**
Unreserve

fffff **GetMaterial**
SetMaterial

, ‘ **GetHandle**
Initialize

IDirect3DMaterial の COM の
IUnknown **IUnknown** の

AddRef
QueryInterface
Release

IDirect3DMaterial::GetHandle

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

Direct3DMaterial の の の の Direct3D API の の 1

- D3D_OK, , , , , DDERR_INVALIDOBJECT,

lpDirect3DDevice
 Ž — , , , , , fffff, Direct3DDevice fffff, ffff

lpHandle
 Direct3DMaterial fffff, ‘%, , , , , ffff, ’, , , , , ffff

IDirect3DMaterial::GetMaterial

`HRESULT GetMaterial(LPD3DMATERIAL lpMat);`

Direct3DMaterial の

- D3D_OK Ž ”, , , Ž ’, , , , , , ,
DDERR_INVALIDOBJECT
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

<i>ffff</i>	GetHandle	
<i>Š%</i>	Initialize	
<i>f f</i>	Load Unload	
<i>ffff •</i>	PaletteChanged	
IDirect3DTexture <i>IUnknown</i>	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

- D3D_OK Ž”,,, Ž,’,,,,,•,
D3DERR_BADOBJECT
DDERR_INVALIDPARAMS

lpDirect3DDevice
f f,,,,fffff,Direct3DDeviceffffff,,ffff
lpHandle
 Direct3DTextureffffff,‘%,,fffff,ffff, ’,,,• ,ffff

IDirect3DTexture::Initialize

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

COM ののの

- Direct3DTextureffffff, Ž, Š%
,,, DDERR_ALREADYINITIALIZED,•,

lpDirect3D
Direct3Dfffff,Ž,Direct3D “,,ffff

lpDDSurface
,,fffff ,DirectDraw,,ffff

IDirect3DTexture::Load

HRESULT Load(LPDIRECT3DTEXTURE lpD3DTexture);

DDSCAPS_ALLOCONLOAD

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

lpD3DTexture
f f,,,fffff,,ffff
IDirect3DTexture::Unload

IDirect3DTexture::PaletteChanged

HRESULT PaletteChanged(DWORD dwStart, DWORD dwCount);

の

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

dwStart
• ,,, “,ffff,ffffff

dwCount
• ,,,ffff,Ē

〇〇〇〇

IDirect3DTexture::Unload

HRESULT Unload();

の

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

IDirect3DTexture::Load

IDirect3DViewport

IDirect3DViewport のののIDirect3DViewport のIDirect3DViewport

IDirect3DViewport のの

”Ⓔ	GetBackground
	GetBackgroundDepth
	SetBackground
	SetBackgroundDepth

Š‰	Initialize
----	------------

ⒺⒺ	AddLight
	DeleteLight
	LightElements
	NextLight

fffff,ff f f	Clear
	GetViewport
	SetViewport

•Š	TransformVertices
----	-------------------

IDirect3DViewport	の COM	の
<i>IUnknown</i>	IUnknown	

AddRef
QueryInterface
Release

IDirect3DViewport::AddLight

HRESULT AddLight(LPDIRECT3DLIGHT lpDirect3DLight);

Direct3DLight の

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

lpDirect3DLight
 Direct3DDeviceffffff,Š~,Direct3DLightffffff,,ffff

IDirect3DViewport::Clear

HRESULT Clear(DWORD dwCount, LPD3DRECT lpRects, DWORD dwFlags);

ののの

- D3D_OK, D3D_ERR_INVALIDOBJECT, D3D_ERR_INVALIDPARAMS

dwCount

lpRects

lpRects

D3DRECT

dwFlags

D3DCLEAR_TARGET, D3DCLEAR_ZBUFFER

D3DCLEAR_ZBUFFER

D3DCLEAR_ZBUFFER

IDirect3DViewport::DeleteLight

HRESULT DeleteLight(LPDIRECT3DLIGHT lpDirect3DLight);

Direct3DLight

- D3D_OK, D3D_ERR_INVALIDOBJECT, D3D_ERR_INVALIDPARAMS

lpDirect3DLight

Direct3DDevice, Direct3DLight

IDirect3DViewport::GetBackground

HRESULT GetBackground(LPD3DMATERIALHANDLE lphMat, LPBOOL lpValid);

- D3D_OK, D3D_ERR_INVALIDOBJECT, D3D_ERR_INVALIDPARAMS

lphMat

D3DMATERIALHANDLE

lpValid

LPBOOL

IDirect3DViewport::SetBackground

IDirect3DViewport::GetBackgroundDepth

HRESULT GetBackgroundDepth(LPDIRECTDRAWSURFACE* lpLpDDSurface,

```
LPBOOL lpValid);
```

DirectDraw

- `E,,D3D_OK` `Ž", Ž',,,,,,;`
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

 $lplpDDSurface$

```
”E “,•,DirectDrawSurfacefffff,,ffff,,ffff
```

 $lpValid$

ff f f,Š•,,,"€ “,,,,FALSE, ',,• ,ffff

IDirect3DViewport::SetBackgroundDepth

IDirect3DViewport::GetViewport

```
HRESULT GetViewport(LPD3DVIEWPORT lpData);
```

- `E,,D3D_OK` `Ž", Ž',,,,,;`
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

 $lpData$

```
ff f f,Ž“,D3DVIEWPORT “,ffff
```

IDirect3DViewport::SetViewport

IDirect3DViewport::Initialize

```
HRESULT Initialize(LPDIRECT3D lpDirect3D);
```

COM の の の

- Direct3DViewportffffff, ž, š%
,,, DDERR_ALREADYINITIALIZED,•

 $lpDirect3D$

Direct3Dfffff,Ž,Direct3D “.,ffff

IDirect3DViewport::LightElements

```
HRESULT LightElements(DWORD dwElementCount, LPD3DLIGHTDATA lpData);
```

IDirect3DViewport::NextLight

```
HRESULT NextLight(LPDIRECT3DLIGHT lpDirect3DLight,  
    LPDIRECT3DLIGHT* lplpDirect3DLight, DWORD dwFlags);
```

Direct3DLight

- `E,,D3D_OK` `Ž", Ž',,,,,,;`
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpDirect3DLight

Direct3DDeviceffffff,Š~,œœfff,',œœ,,ffff

lplpDirect3DLight

```
Direct3DDeviceffffff,Š,,€€fff,',—,€,€€,Š,ffff,,ffff
—,€,€€, dwFlagsfff f,Ž',,,
```

dwFlags

Œœfff,,Ž“,œœ,Ž’,fff fffff,, D3DNEXT_NEXT, ’,,

D3DNEXT_HEAD *fff*, , -D3DNEXT_NEXT *fff,ž, –*D3DNEXT_TAIL *fff, 0, -*

IDirect3DViewport::SetBackground

```
HRESULT SetBackground(D3DMATERIALHANDLE hMat);
```

- `E,,D3D_OK` `Ž` ,, `Ž,` ,, ,, ,, ,, ,, ,, `;`
`DDERR_INVALIDOBJECT`
`DDERR_INVALIDPARAMS`

 $hMat$

"E,,Ž—,,fffff,fffff

IDirect3DViewport::GetBackground

IDirect3DViewport::SetBackgroundDepth

```
HRESULT SetBackgroundDepth(LPDIRECTDRAWSURFACE lpDDSurface);
```

- `E,,D3D_OK` `Ž", Ž',,,,,,•,`
`DDERR_INVALIDOBJECT`
`DDERR_INVALIDPARAMS`

lpDDSurface

”Œ “,•,DirectDrawSurfaceffffff,ffff

IDirect3DViewport::Clear

D3DCLEAR_ZBUFFER

Z

16

IDirect3DViewport::GetBackgroundDepth

IDirect3DViewport::SetViewport

HRESULT SetViewport(LPD3DVIEWPORT lpData);

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

lpData

Œff f f,Ž,D3DVIEWPORT “,ffff

IDirect3DViewport::GetViewport

IDirect3DViewport::TransformVertices

HRESULT TransformVertices(DWORD dwVertexCount,
LPD3DTRANSFORMDATA lpData, DWORD dwFlags, LPDWORD lpOffscreen);

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

dwVertexCount

itfff f,’,•Š,,,”

lpData

•Š,,,”,Š,D3DTRANSFORMDATA “,ffff

dwFlags

^%o,fff,“,,, ,fff,Ž,•,,,, Œ , –,Ž ,,,,

D3DTRANSFORM_CLIPPED

D3DTRANSFORM_UNCLIPPED

lpOffscreen

•ŠŒ ”,fff fŠ,, 0Š,’,’,,,• ,ffff

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
D3DTRANSFORMDATA の drExtent のの

D3DTRANSFORMDATA の dwClip 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,Œ,,, Œ%
,dwValuefff,Ž’,,,‘,, bNegatefff,FALSE, •Š,,
Ž—%”,ffff ‘fff,fff, D3DSTATUS “,dwStatus fff,Ž ,,,

dwValue

dwMaskfff,Ž,,,—,”Š,,,ffff fff’<,’

bNegate

TRUE, ”ŠŒ%”,“

dwOffset

•Š ,,fffff —,,, 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

dvR, dvG, dvB, dvA
 $, -',,, - ffff, \tilde{Z}',, D3DVALUE \mathbb{E},'$

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

,, “, — \mathbb{E} , f f, ’,,,,, fff, \tilde{Z} •,, fff

D3DDD_BCLIPPING

bClipping fff, — \mathbb{E}

D3DDD_COLORMODEL

dcmColorModel fff, — \mathbb{E}

D3DDD_DEVCAPS

dwDevCaps fff, — \mathbb{E}

D3DDD_LIGHTINGCAPS

dlcLightingCaps fff, — \mathbb{E}

D3DDD_LINECAPS

dpcLineCaps fff, — \mathbb{E}

D3DDD_MAXBUFFERSIZE

dwMaxBufferSize fff, — \mathbb{E}

D3DDD_MAXVERTEXCOUNT

dwMaxVertexCount fff, — \mathbb{E}

D3DDD_TRANSFORMCAPS

dtcTransformCaps fff, — \mathbb{E}

D3DDD_TRICAPS

dpcTriCaps fff, — \mathbb{E}

dcmColorModel

ffff,ff fff,Ž',,,,D3DCOLORMODEL—Œfff,,,

dwDevCaps

ffff,“—,Ž•,,fff

D3DDEVCAPS_EXECUTESYSTEMMEMORY

ffff fff,Ž ffff,Ž—%”

D3DDEVCAPS_EXECUTEVIDEOMEMORY

fff fff,Ž ffff,Ž—%”

D3DDEVCAPS_FLOATTILVERTEX

•ŠŒ,“f f,“ “,Ž“

D3DDEVCAPS_SORTDECREASINGZ

“,Œ •Œ,f f,,f f,•—

D3DDEVCAPS_SORTEXTACT

—,f f,,f f,•—

D3DDEVCAPS_SORTINCREASINGZ

Z’%•Œ,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,ŒŒ”—,Ž’,,D3DLIGHTINGCAPS “,fff,,,

dpcLineCaps□dpcTriCaps

,ŽŠŒ,•%,,, ffff,ff f •,’<,,D3DPRIMCAPS “

dwDeviceRenderBitDepth

ffff,ffffff¥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**
 の

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,ffffff

```

dwVertexCount

```

Ž ,,“

```

dwInstructionOffset

```

Ž —fff,ffffff

```

dwInstructionLength

```

Ž —,’

```

dwHVertexOffset

```

ffff fff,ffffff,•—,,fff f •f f,—,,,,,Ž—,,“Ž,’“,ffffff

```

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—Œfff,,,

```

guid

```

Œ ,,ffff,ff ff ff fŽ•Ž(GUID)

```

dwCaps

```

”—,fff

```

dpcPrimCaps

```

,,,,ffffffŒ,,,,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;

```

○D3DTRANSFORMDATA ○**dwFlags**

```

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

```

dvHX dvHY dvHZ

```

“Ž •,•Š,Ž’,D3DVALUEŒ,’ ,,,, •, ’“Ž,

```

D3DINSTRUCTION

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

ののののの

bOpcode

D3DOPCODE — $\langle \mathbb{E}, fff, \dots, \checkmark', \dots, ffffff \rangle$ —

bSize

— $f \ fffff, fff \ , fff, \ \checkmark, \checkmark, \text{---}, \dots, \checkmark, \dots$

wCount

$\prime, \text{---} f \ f \ \ , fff, \ \checkmark \checkmark \mathbb{E}, ffff, \dots, \checkmark \checkmark \mathbb{E}, \dots, \checkmark, \text{---}, \dots, \checkmark, \checkmark, \dots$

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

$\ , \ , \ \langle \prime, fff \rangle^{\wedge}, fff$

dltType

$\mathbb{E} \mathbb{E}, \mathbb{E} \ \ , \prime, \ \text{D3DLIGHTTYPE} \text{---} \langle \mathbb{E}, fff, \dots, \dots, \dots$

dcvColor

$\mathbb{E} \mathbb{E}, \ \ , fff, \ \text{D3DCOLORVALUE} \ \langle \prime, \dots$

dvPosition□**dvDirection**

$f \ ff \langle \checkmark, \ , \mathbb{E} \mathbb{E}, \wedge^{\prime}, \rangle^{\wedge}$

dvRange

$\mathbb{E} \mathbb{E}, \text{---} \mathbb{E}^{\wedge}$

dvFalloff

$\%_0(\text{dvTheta} fff, \checkmark', \dots, \checkmark \langle \prime \rangle), \ \%_0, \checkmark \sim (\text{dvPhi} fff, \checkmark', \dots, \checkmark \langle \prime \rangle)$
 $, \checkmark, \langle \prime \rangle, \mathbb{E} \ \ , \ \ \%_0 \langle \prime \rangle, \langle \prime \rangle, \langle \prime \rangle, \ \checkmark, \bullet \checkmark, \checkmark, \checkmark, \dots$

$$Light \times \cos^{falloff} \left(\frac{\pi}{2} \left(\frac{2\rho - dvTheta}{dvPhi - dvTheta} \right) \right)$$

, , 'Ž , , rho, ŽŽ, fffffff' , , Š , ,
dvAttenuation0
 ^ , < “ Ą Ą , dvRange fff , , — , , , , Ž “ , Š , Ą , , , Ą Ą fff , Ž ' , ,
dvAttenuation1
 fff , Ą , , < “ , dvRange fff , , — , , , , Ž “ , Š , Š , , , 50f fff , ,
dvAttenuation2
 2Ž Ą — , , , Ą , , < “
dvTheta
 fffffff , Š (ffff) , , , fffffff , % Ą Ą , , ,
dvPhi
 fffffff , ŠŠ (ffff) , , % Ą Ą , Š , < “ , fffffff , , , , ,
 3 3 3 d d d

$$attenuation = attenuation_0 + attenuation_1 \times d + attenuation_2 \times d^2$$

3

D3DLIGHTTYPE

D3DLIGHTDATA

```
typedef struct _D3DLIGHTDATA {
    DWORD          dwSize;
    LPD3DLIGHTINGELEMENT lpIn;
    DWORD          dwInSize;
    LPD3DTLVERTEX  lpOut;
    DWORD          dwOutSize;
} D3DLIGHTDATA, *LPD3DLIGHTDATA;
```

IDirect3DViewport::LightElements 3

dwSize
 , , “ , fff ^ , fff

lpIn
 “ — “ , — ffff , Ž , , D3DLIGHTINGELEMENT “ , , ffff

dwInSize
 Ž , “ — “ , , ^ — , , , ffff fff , — “ , fffff — , — Š f f , Š ” , , , , , ^ , , ,

lpOut
 — , Ž , , D3DTLVERTEX “ , , ffff

dwOutSize
 Ž , — , , , ^ — , , , ffff fff , fffff — , — Š f f , Š ” , , , , , ^ , , ,

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,ffffffff,ff f,,

D3DLIGHTCAPS_PARALLELPOINT

• ☒,ff f,,

D3DLIGHTCAPS_POINT

“☒☒,ff f,,

D3DLIGHTCAPS_SPOT

ffffffff,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¥ff 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;
```

```

        D3DCOLOR dcColor;
    };
    union {
        D3DCOLOR specular;
        D3DCOLOR dcSpecular;
    };
    union {
        D3DVALUE tu;
        D3DVALUE dvTU;
    };
    union {
        D3DVALUE tv;
        D3DVALUE dvTV;
    };
} D3DLVERTEX, *LPD3DLVERTEX;

```

(の)ののの

dvX, dvY, dvZ
 ’“,“Ž •,Ž’,,D3DVALUE€,’
dwReserved
 — , 0,,,,,,,,,
dcColorꠔdcSpecular
 ’“, ,“%o,Ž’,,D3DCOLOR€,’
dvTUꠔdvTV
 ’“,fffff •,Ž’,,D3DVALUE€,’

D3DMATERIAL

```

typedef struct _D3DMATERIAL {
    DWORD          dwSize;
    union {
        D3DCOLORVALUE diffuse;
        D3DCOLORVALUE dcvDiffuse;
    };
    union {
        D3DCOLORVALUE ambient;
        D3DCOLORVALUE dcvAmbient;
    };
    union {
        D3DCOLORVALUE specular;
        D3DCOLORVALUE dcvSpecular;
    };
    union {
        D3DCOLORVALUE emissive;
        D3DCOLORVALUE dcvEmissive;
    };
    union {
        D3DVALUE          power;
        D3DVALUE          dvPower;
    };
    D3DTEXTUREHANDLE      hTexture;
    DWORD                  dwRampSize;
};

```

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

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

dwSize

```
,, “,fff”^,fff
```

d3vDiffuse **d3vAmbient** **d3vSpecular** **d3vEmissive**

```
,,, fffff,ŠŽ   •^   “%o   •Ž ,Ž’,,’ ,,,,’, D3DCOLORVALUE “,
```

d3vPower

```
“%o,ffffff,<,,Ž’,,D3DVALUEœ,’
```

hTexture

```
ffffff¥fff,ffff
```

dwRampSize

```
,œŽ,’,’ ,(œŽ)ffff,, ”œ,Ž’,,,,ffffff,,,, ,,’1^%o  
,,,,,,,, ,,’,” ”œ,•Ž,,,, ,,” ”œ,Ž’,,,,ffffff,dwRampSizefff,1^ ,fff  
ff,œ ,,,,,,*,,
```

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**

```
“ ”,“œ, —,fff
```

D3DOPCODE

D3DMATRIXMULTIPLY

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

D3DOPCODE の D3DOP_MATRIXMULTIPLY の¥

hDestMatrix

ŒŽŒ‰,Š”,, —,ffff

hSrcMatrix1 □ hSrcMatrix2

1”–,2”–,ŒŽ‘ , —,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

Œ ,,,“ffffff,Ž ffff, “,,,ffffff

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¥

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;
```

のののののD3DDEVICEDESC ののの

dwSize
 ,, “,fff”^,fff

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

D3DPMISCCAPS_CONFORMANT

ffff, OpenGL• , ,,,

D3DPMISCCAPS_CULLCCW

ffff, D3DRENDERSTATE_CULLMODE ‘,“,, %
 ,,ff f,,(ŽŠŒ,ffffff, ,,) **D3DCULL**
 —ŒŒ,D3DCULL_CCWfff,‘%,,

D3DPMISCCAPS_CULLCW

ffff, D3DRENDERSTATE_CULLMODE ‘,“,,%
 ,,ŽŠŒ,ff f,,(ŽŠŒ,ffffff, ,,) **D3DCULL**
 —ŒŒ,D3DCULL_CCWfff,‘%,,

D3DPMISCCAPS_CULLNONE

ffff, “,,,ŽŠŒ,Ž ,,, **D3DCULL**—ŒŒ,D3DCULL_NONEfff,‘%
 ,,

D3DPMISCCAPS_LINEPATTERNREP

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

D3DPMISCCAPS_MASKPLANES

ffff, ,ffffff,Ž ,,,
D3DPMISCCAPS_MASKZ
ffff, ffff —,Zffff, ,%o”,,,•%o”,,,,,

dwRasterCaps

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

D3DPRASTERCAPS_ROP2

ffff, R2_COPYPEN^Š,fff —,ff f,,

D3DPRASTERCAPS_STIPPLE

ffff, ”“—,ffff f,,ffff,“•,,,

D3DPRASTERCAPS_SUBPIXEL

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

D3DPRASTERCAPS_SUBPIXELX

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

D3DPRASTERCAPS_XOR

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

D3DPRASTERCAPS_ZTEST

ffff,Zfff —
,Ž ,,, ,,, Ć%o“,ffffff,ffffff, fffffff,,,,,,Zffff,,•Ž,,

dwZCmpCaps

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

D3DPCMPCAPS_ALWAYS

,,,Zfff,’

D3DPCMPCAPS_EQUAL

Ć,Z,Ć ,Z,“,, Zfff,’

D3DPCMPCAPS_GREATER

$$\langle Z, \mathbb{E} \rangle, Z, ', , Zfff, ',$$
D3DPCMPCAPS_GREATEREQUAL

$$\langle Z, \mathbb{E} \rangle, Z, ', , , , Zfff, ',$$
D3DPCMPCAPS_LESS

$$\langle Z, \mathbb{E} \rangle, Z, ', , Zfff, ',$$
D3DPCMPCAPS_LESSEQUAL

$$\langle Z, \mathbb{E} \rangle, Z, ', , , , Zfff, ',$$
D3DPCMPCAPS_NEVER

$$', Zfff, \checkmark, ',$$
D3DPCMPCAPS_NOTEQUAL

$$\langle Z, \mathbb{E} \rangle, Z, ', , , Zfff, ',$$
dwSrcBlendCaps

$$f \ f \ \bullet \ , fff, \checkmark, ', , ^ \ , , (RGBA', f \ f, fffff \ fff, ', , s, d, \bullet,)$$
D3DPBLENDCAPS_BOTHINVSRCALPHA

$$f \ f \ \text{---} '(1-A_s, 1-A_s, 1-A_s, 1-A_s), fffff \ fff \ \text{---} '(A_s, A_s, A_s, A_s) \ fffff \ fff \ , ', \text{---} \mathbb{E}, , ,$$
D3DPBLENDCAPS_BOTHSRCALPHA

$$f \ f \ \text{---} '(A_s, A_s, A_s, A_s), fffff \ fff \ \text{---} '(1-A_s, 1-A_s, 1-A_s, 1-A_s) \ fffff \ fff \ , ', \text{---} \mathbb{E}, , ,$$
D3DPBLENDCAPS_DESTALPHA

$$\text{---} '(A_d, A_d, A_d, A_d)$$
D3DPBLENDCAPS_DESTCOLOR

$$\text{---} '(R_d, G_d, B_d, A_d)$$
D3DPBLENDCAPS_INVDESTALPHA

$$\text{---} '(1-A_d, 1-A_d, 1-A_d, 1-A_d)$$
D3DPBLENDCAPS_INVDESTCOLOR

$$\text{---} '(1-R_d, 1-G_d, 1-B_d, 1-A_d)$$
D3DPBLENDCAPS_INVSRCALPHA

$$\text{---} '(1-A_s, 1-A_s, 1-A_s, 1-A_s)$$
D3DPBLENDCAPS_INVSRCOLOR

$$\text{---} '(1-R_d, 1-G_d, 1-B_d, 1-A_d)$$
D3DPBLENDCAPS_ONE

$$\text{---} '(1, 1, 1, 1)$$
D3DPBLENDCAPS_SRCALPHA

$$\text{---} '(A_s, A_s, A_s, A_s)$$
D3DPBLENDCAPS_SRCALPHASAT

$$\text{---} '(f, f, f, 1) \ f = \min(A_s, 1-A_d)$$
D3DPBLENDCAPS_SRCCOLOR

$$\text{---} '(R_s, G_s, B_s, A_s)$$

D3DPBLENDCAPS_ZERO

—(0, 0, 0, 0)

dwDestBlendCaps

fffff 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',,,,^ ' ,,,

D3DPSHADECAPS_ALPHAFLATBLEND

D3DPSHADECAPS_ALPHAFLATSTIPPLED

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

D3DPSHADECAPS_ALPHAGOURAUDBLEND

D3DPSHADECAPS_ALPHAGOURAUDSTIPPLED

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

D3DPSHADECAPS_ALPHAPHONGBLEND

D3DPSHADECAPS_ALPHAPHONGSTIPPLED

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

D3DPSHADECAPS_COLORFLATMONO

D3DPSHADECAPS_COLORFLATRGB

ffff, D3DCOLOR_MONO,D3DCOLOR_RGBff f f,•— ff ffff, •,ff f,, ,,,f f,, ,,,—‘, •, —‘, ,’“, ,,,— ,,, ffff(Ef f, , •,, ,Š,,, RGBEf f,, — •, ,Š,,,

D3DPSHADECAPS_COLORGOURAUDMONO

D3DPSHADECAPS_COLORGOURAUDRGB

ffff, D3DCOLOR_MONO,D3DCOLOR_RGBff f f,ff ff f fff, •,ff f,, ,,,f f, ,,,—‘,,,ffff •,’“,,,,, ‘, •,,,•—

“%00ffffff,ff f,,(€ ,“%00 ,,,,,,ffffff,•%00,,)

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_LINEARMIPLINEAR

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

D3DPTFILTERCAPS_MIPLINEAR

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

D3DPTFILTERCAPS_MIP_NEAREST

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',,,,^ ',,,

D3DPTBLEND CAPS_COPY

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

D3DPTBLEND CAPS_DECAL

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

D3DPTBLEND CAPS_DECALALPHA

ff f¥ffff¥ffffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_DECALALPHA),ff f,,

D3DPTBLEND CAPS_DECALMASK

ff f¥ffff¥ffffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_DECALMASK),ff f,,

D3DPTBLEND CAPS_MODULATE

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

D3DPTBLEND CAPS_MODULATEALPHA

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

D3DPTBLEND CAPS_MODULATEMASK

•'ffff¥ffffff f f(D3DTEXTUREBLEND
—<E,,D3DPTBLEND_MODULATEMASK),ff f,,

dwTextureAddressCaps

fffff,fffff • ,fff, Ž',,,,^ ',,,

D3DPTADDRESSCAPS_CLAMP

ffff, fffff,CE',%o",,,

D3DPTADDRESSCAPS_MIRROR

ffff, fffff,"",%o",,,

D3DPTADDRESSCAPS_WRAP

ffff, fffff,'*,%o",,,

dwStippleWidth □ dwStippleHeight

ff f,,“, ‘, ,(32 32^)

D3DPROCESSVERTICES

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

のD3DOPCODE のD3DOP_PROCESSVERTICES**dwFlags**

ffff,,“, —,Ž',Ž,fff, ,,,^ ',,

D3DPROCESSVERTICES_COPY

“ *, ffff, —,, Ž ffff,'“ *, —,, ffff,“, —,•—
,,,, ,CE%o —“,%o',,,

D3DPROCESSVERTICES_NOCOLOR

“, •,.,.,.

D3DPROCESSVERTICES_OPMASK

D3DPROCESSVERTICES_NOCOLOR,

D3DPROCESSVERTICES_UPDATEEXTENTS,Ž ,dwFlagsfff,‘,f
ff,ffffff,Ž',,

D3DPROCESSVERTICES_TRANSFORM

“,Š,,

D3DPROCESSVERTICES_TRANSFORMLIGHT

“,Š,, -CE%o,—,,

D3DPROCESSVERTICES_UPDATEEXTENTS

,,•ŠCE,“, ”^ ,, •,D3DSTATUS “,drExtentfff,•,

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

D3DSTATE

```
typedef struct _D3DSTATE {
    union {
        D3DTRANSFORMSTATETYPE dtstTransformStateType;
        D3DLIGHTSTATETYPE      dlstLightStateType;
        D3DRENDERSTATETYPE      drstRenderStateType;
    };
    union {
        DWORD                    dwArg[1];
        D3DVALUE                 dvArg[1];
    };
} D3DSTATE, *LPD3DSTATE;
```

D3DOPCODE の D3DOP_STATETRANSFORM

D3DOP_STATELIGHT D3DOP_STATEENDER の の の 2 の の の

dtstTransformStateType dlstLightStateType drstRenderStateType
 fffffff 'Ž',,,D3DTRANSFORMSTATETYPE D3DLIGHTSTATETY
 PE D3DRENDERSTATETYPE—Œ,fff,,,

dvArg

,, “, ,fff,Ž',,,Œ,'

D3DLIGHTSTATETYPE D3DOPCODE
D3DRENDERSTATETYPE D3DTRANSFORMSTATETYPE D3DVALUE

D3DSTATS

```
typedef struct _D3DSTATS {
    DWORD dwSize;
    DWORD dwTrianglesDrawn;
    DWORD dwLinesDrawn;
    DWORD dwPointsDrawn;
    DWORD dwSpansDrawn;
    DWORD dwVerticesProcessed;
} D3DSTATS, *LPD3DSTATS;
```

IDirect3DDevice::GetStats

dwSize

,, “,fff^,fff

dwTrianglesDrawn, dwLinesDrawn, dwPointsDrawn, dwSpansDrawn

ffff, ^Œ•%o,,ŽŠŒ “ fff,

dwVerticesProcessed

ffff, ^Œ —,,,’“,

IDirect3DDevice::GetStats

D3DSTATUS

```
typedef struct _D3DSTATUS {
    DWORD dwFlags;
    DWORD dwStatus;
```

```

        D3DRECT drExtent;
    } D3DSTATUS, *LPD3DSTATUS;

```

ののD3DEXECUTEDATA の D3DOPCODE のの D3DOP_SETSTATUS

dwFlags

```

    ff ff ”^ ,,,,—•,,,,,Ž’,,Ž,fff,,,, ’,,

```

D3DSETSTATUS_STATUS

```

    ff ff, ’

```

D3DSETSTATUS_EXTENTS

```

    drExtentfff,Ž’,,”^, ’

```

D3DSETSTATUS_ALL

```

    ff ff,”^,—•, ’

```

dwStatus

```

    ffffffff¥fff ,,fff, Ž,fff,,,^ , ’,,

```

```

    ‘, ,,”fff

```

D3DSTATUS_CLIPINTERSECTION

```

    ,,,,CLIPINTERSECTIONfff,‘, ,,

```

D3DSTATUS_CLIPUNIONALL

```

    ,,,,CLIPUNIONfff,‘, ,,

```

D3DSTATUS_DEFAULT

```

    D3DSTATUS_CLIPINTERSECTION,

```

```

    D3DSTATUS_ZNOTVISIBLEfff,‘, , ,’, fffff,,

```

D3DSTATUS_ZNOTVISIBLE

```

    fffffŒ fff

```

D3DSTATUS_CLIPINTERSECTIONBACK

```

    %0 Œ,Œ•ffff–,’“,ffffff fff,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONBOTTOM

```

    %0 Œ,%0,’“,ffffff fff,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONFRONT

```

    %0 Œ,‘•ffff–,’“,ffffff fff,~— ,,,

```

**D3DSTATUS_CLIPINTERSECTIONGEN0 D3DSTATUS_CLIPINTERSECTI
ONGEN5**

```

    ffff fff’Œ,ffff–,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONLEFT

```

    %0 Œ, ‘,’“,ffffff fff,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONRIGHT

```

    %0 Œ,%0,’“,ffffff fff,~— ,,,

```

D3DSTATUS_CLIPINTERSECTIONTOP

```

    %0 Œ, ,’“,ffffff fff,~— ,,,

```



```

ffff& 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 &,”-,,,ffffff,,,
D3DCLIP_BOTTOM
    %0 &,’-,,,ffffff,,,
D3DCLIP_FRONT
    %0 &,’-,,,ffffff,,,
D3DCLIP_LEFT
    %0 &,’-,,,ffffff,,,
D3DCLIP_RIGHT
    %0 &,%0’-,,,ffffff,,,
D3DCLIP_TOP
    %0 &,’-,,,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

“œfffff,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’,,D3DVALUEœ,’

dvRHW

D3DVALUE €, ' , , , , ZŽ, %, , € “ , , fffffff, , ←, Š, , , ,

dcColor □ **dcSpecular**

“ , , “ %, Ž , , **D3DCOLOR** €, ' ,

dvTU □ **dvTV**

“ , , fffff Ž , , **D3DVALUE** €, ' ,

D3DLIGHTDATA

D3DTRANSFORMCAPS

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

の **D3DDEVICEDESC** の

dwSize

“ , , “ , fff” ^, fff

dwCaps

• Š , , ffff, fffffff, , , , , Ž , , fff , , fff, 0, , , Ž, fff, ' , , ,

D3DTRANSFORMCAPS_CLIP ffff, • Š , , fffffff, ,

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 ,,’, D3DVERTEX “,,

dwClip

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

D3DCLIP_BACK

%0 E,”-,,,ffffff,,

D3DCLIP_BOTTOM

%0 E,’-,,,ffffff,,

D3DCLIP_FRONT

%0 E,‘-,,,ffffff,,

D3DCLIP_GEN0 through D3DCLIP_GEN5

ffff fff’\,fff-

D3DCLIP_LEFT

%0 E, ‘-,,,ffffff,,

D3DCLIP_RIGHT

%0 E,%0‘-,,,ffffff,,

D3DCLIP_TOP

%0 E, ’-,,,ffffff,,

dwClipIntersection

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

D3DSTATUS_CLIPINTERSECTIONBACK

%0 E,E•ffff-,’“,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONBOTTOM

%0 E,%0,’“,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONFRONT

%0 E,‘•ffff-,’“,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONGEN0 D3DSTATUS_CLIPINTERSECTIONGEN5

ffff fff’\,fff-,~— ,,,

D3DSTATUS_CLIPINTERSECTIONLEFT

%0 E, ‘,’“,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONRIGHT

%0 E,%0‘,’“,ffffff fff,~— ,,,

D3DSTATUS_CLIPINTERSECTIONTOP

%0 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

```

•ŠĖ,“,”^,Ė’,’,’, “, fffff•ŠĖ,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
ŽŠĖ,Ž’,’,“

```

wFlags
 Ž—%”,ŽŠŒ,fff,’ċ,,fff(,, •, fffff f f f,,—
 (Œ) ,,fff,, Ž,’,,,^ ’,,
fff¥fff
D3DTRIFLAG_EDGEENABLE1
 fffv1–v2,’ċ,,
D3DTRIFLAG_EDGEENABLE2
 fffv2–v3,’ċ,,
D3DTRIFLAG_EDGEENABLE3
 fffv3–v1,’ċ,,
D3DTRIFLAG_EDGEENABLETRIANGLE
 ,,,fff,’ċ,,
“Š , Œfff
D3DTRIFLAG_EVEN
 Œ ,ŽŠŒ,v1–v2fff, ’‘,ŽŠŒ,v3–
 v1fff, ,,,, v1,’‘,v1, v2,’‘,v3,,,
D3DTRIFLAG_ODD
 Œ ,ŽŠŒ,v1–v2fff, ’‘,ŽŠŒ,v2–
 v3fff, ,,,, v1,’‘,v3, v2,’‘,v2,,,
D3DTRIFLAG_START
 “Š , Œ,ŠŽ,, ,,”“,f f,,
D3DTRIFLAG_STARTFLAT(len)
 ,,ŽŠŒ, ,”,,,, ,Œ,ŽŠŒ,Ž’, , ,” , ,’ , 0,,‘,30, , ,
 ののののの v1 v2 v3 のののの

The D3DTRIFLAG_ODD D3DTRIFLAG_EVEN
 のののの 5 ののの
 D3DTRIFLAG_START
 D3DTRIFLAG_ODD
 D3DTRIFLAG_EVEN
 D3DTRIFLAG_ODD
 D3DTRIFLAG_EVEN
 の 5 のの
 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;

```

(㉑)

D3DOPCODE の D3DOP_TRIANGLE

```

dvX  dvY  dvZ
<Ž, •,“,Ž’,,D3DVALUE€,’
dvNX dvNY      dvNZ
< •,“,Ž’,,D3DVALUE €,’
dvTU □ dvTV
“,fffff,Ž’,,D3DVALUE€,’
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 \square **dwY**
 ff f f, <, •
dwWidth \square **dwHeight**
 ff f f,”—
dvScaleX \square **dvScaleY**
 fff f,<Ž,ff f”,Ž’,,D3DVALUEŒ,’
dvMaxX **dvMaxY** **dvMinZ** **dvMaxZ**
 x y z,<Ž •, “, ’,Ž’,,D3DVALUEŒ,’
D3DVALUE **IDirect3DViewport::GetViewport**
IDirect3DViewport::SetViewport

—<Œ

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 **D3DRENDERSTATE_DESTBLEND** の
 の **RGBA** sŒd

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_INVSRCALPHA

—‘, (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 , -œ,,

D3DBLEND_BOTHINVSRCALPHA

f f —‘, (1-As, 1-As, 1-As, 1-As) fffff fff —‘, (As, As, As, As) fffff fff , -œ,,

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,Ž”,,

D3DCMP_LESS

,,,fff’, œ ,fff’,,, ,,,,%,,,

D3DCMP_EQUAL

,,,fff’, œ ,fff’,“,,,,%,,,

```

D3DCMP_LESSEQUAL
    „ffff’, Ē ,ffff’^%o,,,%o,,
D3DCMP_GREATER
    „ffff’, Ē ,ffff’,‘,,,,,%o,,
D3DCMP_NOTEQUAL
    „ffff’, Ē ,ffff’“,,,,,,%o,,
D3DCMP_GREATEREQUAL
    „ffff’, Ē ,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‘,RGBfff,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

fff %o , , ,

D3DFOG_EXP

Ž , • Ž , ‘ , fff %o , ‘ ,

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

D3DFOG_EXP2

Ž , • Ž , ‘ 2 , fff %o , ‘ ,

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

D3DFOG_LINEAR

Ž , • Ž , Ž “ , “ , Š , Ǝ , fff %o , ‘ ,

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

„ , Ǝ 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

ⒺⒺ,‘ Ⓔ,,,

D3DLIGHT_PARALLELPPOINT

ⒸⒸ,• ,“,,

D3DLIGHT_GLSPOT

ⒸⒸ,GLⒸ,ffffff,,

IDirect3DRMLight □ *IDirect3DRMLightArray* Interfaces

D3DOPCODE

```
typedef enum _D3DOPCODE {  
    D3DOP_POINT          = 1,  
    D3DOP_LINE           = 2,  
    D3DOP_TRIANGLE       = 3,  
    D3DOP_MATRIXLOAD     = 4,  
    D3DOP_MATRIXMULTIPLY = 5,  
    D3DOP_STATETRANSFORM = 6,  
    D3DOP_STATELIGHT     = 7,  
    D3DOP_STATERENDER    = 8,  
    D3DOP_PROCESSVERTICES = 9,  
    D3DOP_TEXTURELOAD    = 10,  
    D3DOP_EXIT            = 11,  
    D3DOP_BRANCHFORWARD  = 12,  
    D3DOP_SPAN            = 13,  
    D3DOP_SETSTATUS      = 14,  
} D3DOPCODE;
```

⑦

D3DOP_POINT

ffff ,“,’ fffff f f, D3DPOINT “,Ž’,,

D3DOP_LINE

ffff , ,’, fffff f f, D3DLINE “,Ž’,,

D3DOP_TRIANGLE

ffff ,ŽŠⒸ,’, fffff f f,D3DTRIANGLE “,Ž’,,

D3DOP_MATRIXLOAD

ffffff ffff,f f“,,fff ,, fffff f f,D3DMATRIXLOAD “,Ž’,,
,

D3DOP_MATRIXMULTIPLY

ffffff ffff,f f“,,fff ,, fffff f f,D3DMATRIXMULTIPLY ‘
,Ž’,,

D3DOP_STATETRANSFORM

•Šfff f—
,ffffff ffff, “• ‘• ’, ’, fffff f f, • ,,f ff, ,,’, f ff,
“• ‘• ,Ž, ,,• ’, ,, ’, ’, ,,,• ,,,, , D3DSTATE “,D3DTR
ANSFORMSTATETYPE—Ⓒ,Ž ,,,

D3DOP_STATELIGHT

ⒸⒸfff f—
,ffffff ffff,“• ‘• ’, ’, fffff f f, • ,,f ff, ,,’, f ff, “
• ‘• ,Ž, ,,• ’, ,, ’, ’, ,,,• ,,,, , D3DSTATE “,D3DLIGH
TSTATETYPE—Ⓒ,Ž ,,,

D3DOP_STATERENDER

```

ffffffffff 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

```

ffffff 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 ,,,

```

```

() /()()()()()

```

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

TRUE,,,ffffff,%”,,, ffff’,FALSE,,,

D3DRENDERSTATE_BLENDENABLE

TRUE,,,ffff, %”,,, ffff’,FALSE,,,

D3DRENDERSTATE_FOGENABLE

TRUE,,,fff,%”,,, ffff’,FALSE,,,

D3DRENDERSTATE_SPECULARENABLE

TRUE,,,“%o,%o”,,, ffff’,TRUE,,,

D3DRENDERSTATE_ZVISIBLETRUE,,,ZŒ %o”,,, ffff’,FALSE,,, ZŒ ,, •%o
,,ffff,Œ,,,,Zffff,‘,,fff,,, ,,,ffff,ff f,fff f<Š,•,,,ffff,, ,”<,,,,,**D3DRENDERSTATE_SUBPIXEL**

TRUE,,,ffffff,’ %o”,,, ffff’,FALSE,,,

D3DRENDERSTATE_SUBPIXELX

TRUE,,,X,,,’ %o”,,, ffff’,FALSE,,,

D3DRENDERSTATE_STIPPLEDALPHA

TRUE,,,“•,,,ffff,%o”,,, ffff’,FALSE,,,

D3DRENDERSTATE_FOGCOLOR

D3DCOLORŒ,’ ffff’,0,,,

D3DRENDERSTATE_FOGTABLEMODE

D3DFOGMODE—Œ,fff,,, ffff’,D3DFOG_NONE,,,

D3DRENDERSTATE_FOGTABLESTART

fff f ff,ŠŽ ,,, Œfff f f,,ŠŽ,,,fffŒ%o,^,,,

D3DRENDERSTATE_FOGTABLEEND

fff f ff, — ,,, Œfff f f, ‘–“,,fffŒ%o,^,,,

D3DRENDERSTATE_FOGTABLEDENSITY

Œfff f f, ‘fff–“ , ’, ,’”^ 0,1,,,

D3DRENDERSTATE_STIPPLEENABLE

ffff ffff,“•,%o”,,, “•,,,ffff,%o”,,, Œ ,“ff f,–Œ,,,,,,,

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 の 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**ffffff
‘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

$ff\ f\ fffff\ f\ f,ff\ f,,, \text{,,}f\ f,,\ fffff,RGB,ffff',\ fffff,\checkmark-$
 $,,,,,\text{'}\checkmark,$

D3DTBLEND_MODULATE

$\bullet' fffff\ f\ f,ff\ f,,, \text{,,}f\ f,,\ fffff,RGB',\ fffff,\checkmark-$
 $,,,,,RGB',\text{E},\bullet,\ fffff,ffff',\ fffff,\checkmark-,,,,,,\text{'}\checkmark',\text{'}\checkmark,$

D3DTBLEND_DECALALPHA

$ff\ f\ ffff\ fffff\ f\ f,ff\ f,,, \text{,,}f\ f,,\ fffff,RGB,ffff',\ \checkmark,\bullet'\checkmark,$
 $ffff,\checkmark-,,,,,,\text{'}\checkmark,\text{'}\checkmark,$

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

$,,\bullet'\checkmark,\ C,\ A,ffff\ t,ffff\ o,\text{E},ffffff(\text{'}\checkmark),\bullet,$
 $D3DTBLEND_DECALALPHAf\ f,\ fffff,ffff',\ fffff,\checkmark-$
 $,,,,,\text{'}\checkmark',\text{'}\checkmark,$

D3DTBLEND_MODULATEALPHA

$\bullet'ffff,ffffff\ f\ f,ff\ f,,, \text{,,}f\ f,\ fffff,RGB',\ RGB',\text{'},\ fffff,ff$
 $ff',\text{'},$

D3DTBLEND_DECALMASK

$ff\ f\ fff,ffffff\ f\ f,ff\ f,,,$

D3DTBLEND_MODULATEMASK

$\bullet'fff,ffffff\ f\ f,ff\ f,,,$

D3DTBLEND_COPY

$\bullet'\checkmarkffff\ f\ f,ff\ f,,,$

の0 1 の の の の の (1)
(1 1 1) の の

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

$-\text{,,},ffff',\ \text{'},\ \bullet,ffff,\checkmark-,,,, \checkmark',\ \text{'},-\bullet,-$
 $,,, \checkmark',\ \text{'},ff\ f,,,,,,, -\bullet,ff\ f,,,,,,,$

D3DFILTER_LINEAR

$2\ 2,-\text{^},\ \text{'},\text{'},\text{'},ffff\checkmark,-\text{,,},ffff,\checkmark-,,,, \checkmark',\ \text{'},-\bullet,-$
 $,,, \checkmark',\ \text{'},ff\ f,,,,,,, -\bullet,ff\ f,,,,,,,$

D3DFILTER_MIPNEAREST

D3DFILTER_NEAREST, “—, “ffffŒ—,ffffff,Ž—,,

D3DFILTER_MIPLINEAR

D3DFILTER_LINEAR, “—,, “ffffŒ—,ffffff,Ž—,,

D3DFILTER_LINEAR_MIPNEAREST

D3DFILTER_MIPNEAREST, “—,, ,,,Žffffff, ‘,,

D3DFILTER_LINEAR_MIPLINEAR

D3DFILTER_MIPLINEAR, “—,, ,,,Žffffff, ‘,,

D3DTRANSFORMSTATETYPE

```
typedef enum _D3DTRANSFORMSTATETYPE {
    D3DTRANSFORMSTATE_WORLD       = 1,
    D3DTRANSFORMSTATE_VIEW        = 2,
    D3DTRANSFORMSTATE_PROJECTION  = 3,
} D3DTRANSFORMSTATETYPE;
```

D3DOPCODE の **D3DOP_STATETRANSFORM** の **D3DSTATE** の

D3DTRANSFORMSTATE_WORLD

D3DTRANSFORMSTATE_VIEW

D3DTRANSFORMSTATE_PROJECTION

•Š,,f ff ff “%o,”—,’<,, fffff’, NULL,,,(“^,”—)

D3DOPCODE D3DRENDERSTATETYPE

,,‘,Œ

D3DCOLOR

```
typedef DWORD D3DCOLOR, D3DCOLOR, *LPD3DCOLOR;
```

の Direct3D の

D3DRGB D3DRGBA

D3DVALUE

```
typedef float D3DVALUE, *LPD3DVALUE;
```

の Direct3D の

●’

の Direct3D のののの

D3D_OK

D3DERR_BADMAJORVERSION

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