

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



# DirectDraw

- DirectDraw,,,,,.....
- DirectDrawf□ffff.....
- DirectDrawŠ—.....
  - DirectDraw.....
  - „,DirectDraw,“.....
  - DirectDraw HAL.....
  - DirectDraw HEL.....
  - DirectDrawffffff,fff.....
  - ,fff.....
  - 3Df□fff,ff□f.....
  - DirectDraw,Direct3D,“□.....
  - f□fXffffff□f□f.....
  - fff□ff□fff.....
- DirectDrawfff ffffŠ—.....
  - IDirectDraw2fff ffff.....
  - IDirectDrawClipperfff ffff.....
  - IDirectDrawPalette fff ffff.....
  - IDirectDrawSurface2fff ffff.....
- DirectDraw ff ffff.....
  - Tutorial 1: DirectDraw,Š‘.....
  - Tutorial 2: fffffff,fff□ffff,“.....
  - Tutorial 3: fffff□f□f□fff,,,ffff“.....
  - Tutorial 4: ff□¥f□,fffffffffff□fff.....
  - Tutorial 5: ffff,““•□.....
  - „, DirectDrawffff.....
  - “%,ffffff.....
- DirectDrawffffff.....
  - Š□.....
  - f ffffŠ□.....
  - IDirectDraw2.....
  - IDirectDrawClipper.....
  - IDirectDrawPalette.....
  - IDirectDrawSurface2.....

---

“ .....  
•’ .....

# DirectDraw,,,,

DirectDraw® の DirectX™ SDK の  
DirectDraw の Microsoft® Windows® のの

DirectDraw Windows ¥ ¥ (GDI)  
のののの¥ ¥ (API) 3D  
Windows のの

DirectDraw ¥の SVGA  
RGB の DirectDraw  
DirectX

DirectDraw のの

- ffffffff,fffffff ffffffff,ff f
- ffffffff¥f f,f ffff ffff“,,ffff, ☒
- 3D Zffff ,ff f
- Zf ffff,f ffff f f ff,ff f
- ff f 'f ffff,,ffff
- ffff f,,fffffff‘,fffffff¥fff—^,“Zffff

DirectDraw の¥のDirectDraw ¥  
RGB YUV  
DirectDraw ののの  
¥

Windows 95 Windows NT® version 4.0  
DirectDraw の

# DirectDrawf □ ffffff

DirectDraw ¥ ¥

DirectDraw  
DirectDraw の (HAL)  
¥

# DirectDrawŠ—

のDirectDraw の

- *DirectDraw*
- *,,’, DirectDraw,“’*
- *DirectDraw HAL*
- *DirectDraw HEL*
- *DirectDrawffffff,fff*
- *,fff*
- *3D f fff,ff f*
- *DirectDraw,Direct3D,“*
- *f fXffffff f f*
- *ffff ff fff*

## DirectDraw

DirectDraw

DirectDraw

¥  
DirectDraw のDirectDraw DirectDraw  
(HAL)のの DirectDraw  
HAL

DirectDraw Ddraw ¥ ¥ (DLL)  
32 DLL DirectDraw の  
DLL の Win32® HAL 16 の  
¥ ¥ DirectDraw のの¥の¥ ¥  
COM

DirectDraw *IDirectDraw2* *IDirectDrawSurface2* *IDirectDrawPalette*  
*IDirectDrawClipper* 4 COM COM  
の ¥ ¥

DirectDraw の DirectDraw DirectDrawSurface DirectDraw  
**DirectDrawCreate** ¥ ¥  
*IDirectDraw2* DirectDrawSurface  
¥

**IDirectDraw2::CreateSurface**

¥のの  
¥ ¥ ¥  
¥ ¥ DirectDrawSurface  
¥

**IDirectDrawSurface2::Flip**

の  
 ¥  
 ¥の  
**IDirectDrawSurface2::Flip**  
 Windows 3.1 GDI ¥  
 DirectDraw  
 のの¥ の ¥の Windows 95 GDI DirectDraw  
 Windows NT  
**CreatedDIBSection**  
 GDI DirectDraw  
 の DirectDraw DirectDrawPalette ¥  
 ¥  
**IDirectDrawPalette** の  
 Windows ¥  
 DirectDrawPalette DirectDraw  
 DirectDraw DirectDrawClipper DirectDraw  
 ¥の

„DirectDraw,“

DirectDraw DirectDraw

¥  
 ¥  
 ¥  
 DirectDraw ¥  
 のの  
 のの  
 DirectDrawSurface ¥ ¥  
 ¥  
 Z

DirectDraw HAL

DirectDraw ¥ (HAL) の  
 HAL 16 32 Windows 95 のWindows NT  
 32 HAL ¥ ¥  
 ¥  
 DLL

---

DirectDraw HAL HAL  
の  
HAL HAL  
DirectDraw

## DirectDraw HEL

DirectDraw ¥ ¥ (HEL) HAL  
DirectDraw DirectDraw HAL  
DirectDraw

## DirectDrawffff,fff

DirectDraw  
DirectDraw DirectDraw

DirectDrawSurface ¥ ¥ ¥  
のの¥の )  
¥の  
DirectDraw ¥

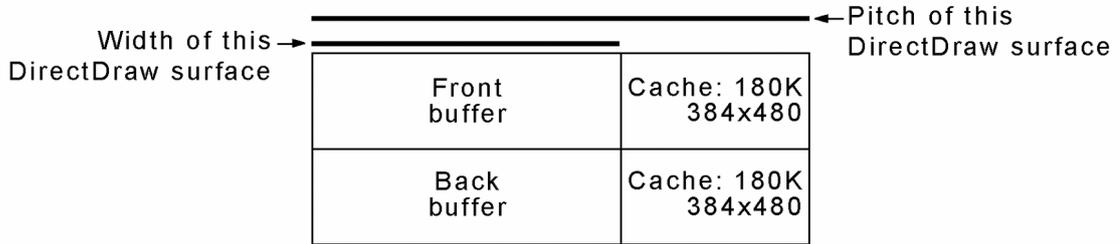
DirectDrawPalette 16 256 の¥

DirectDraw DirectDrawSurface DirectDrawPalette  
DirectDrawClipper の DirectDrawPalette DirectDrawClipper  
DirectDrawSurface  
DirectDrawSurface DirectDrawPalette  
の

## •,fff

¥ のののの  
□□□□ のののの  
のののののののののののののの

¥のののの



の ¥ 640×480×8 384×480×8 の  
640 384 1024

**IDirectDrawSurface2::Lock** ( **IDirectDrawSurface2::GetDC**  
) ¥

¥

**3Df** □ **fff,ff** □ **f**

の DirectDraw 3D

- fffff fff
- ffffff
- Zffff

**ffffff¥fff**

□□□□¥□□□ HEL □  
□□□□¥□□□¥

**DDSURFACEDESC** の **IDirectDraw2::CreateSurface**  
**DDSCAPS\_TEXTURE** ddsCaps

HEL ¥ ¥のの

**ffffff**

DirectDraw ¥ ¥  
のの

の

の

DirectDraw  
のの



```

}
if ((ddres != DD_OK) && (ddres != DDERR_NOTFOUND))
.
.
.

```

¥の¥ ¥  
 ¥の ¥  
 の DDSCAPS\_MIPMAP  
 DDSCAPS\_TEXTURE  
 )

¥ DDSCAPS\_MIPMAP ¥  
 ¥ DDSCAPS\_BACKBUFFER の

のの IDirectDraw2::CreateSurface ¥

IDirectDrawSurface2::AddAttachedSurface ¥  
 ¥

IDirectDrawSurface2::AddAttachedSurface

---

**Note** ffff““ , fffffff ff f,^fff,,“—  
 ,, fffffff,'ff f,ffff“,,,,, Šfff,• ,ffff“,,,,,,,

---

IDirectDrawSurface2::Flip のの  
 □□□□□ □□□□□ ののの  
 ¥ ¥ □□□□□ ¥  
 のの¥ののの¥ □□□□□

IDirectDrawSurface2::Lock IDirectDrawSurface2::GetSurfaceDesc  
 )

DDSURFACEDESC

dwMipMapCount  
 dwMipMapCount

### Zffff

DirectDraw HEL Direct3D™ 3D ¥ Z  
 HEL 16 32 の Z  
 3D の DirectDraw ¥  
 DDSCAPS\_ZBUFFER ¥  
 Z の ¥ DDCAPS の

IDirectDrawSurface2::Blt Z  
 DDBLT\_DEPTHFILL

Z の  
 IDirectDrawSurface2::Blt DDBLTFX Z  
 dwFillDepth 3D の

DirectDraw の Z の DDCAPS\_BLTDEPTHFILL の DDBLT\_DEPTHFILL の Z

**Note** %o ,',Ž ,%oŽ, 3Dffffff ffff,^',,

## DirectDraw, Direct3D, “□

の DirectDraw Direct3D

- Direct3D ffff fff ffff
- Direct3D ffff fff ffff
- Direct3D fffff fff ffff
- DirectDraw HEL, Direct3D

## Direct3D ffff □ fff □ ffff

DirectDraw

DirectDraw Direct3D の Direct3D の COM

DirectDraw の COM

Direct3D DirectDraw COM QueryInterface

Direct3D Direct3D

の DirectDraw

Direct3D

```
LPDIRECTDRAW lpDD;
LPDIRECT3D lpD3D;
ddres = DirectDrawCreate(NULL, &lpDD, NULL);
if (FAILED(ddres))
.
.
.
ddres = lpDD->QueryInterface(IID_IDirect3D,
&lpD3D);
if (FAILED(ddres))
.
.
.
```

の の の の の

**IDirectDraw2::QueryInterface**

2 の Direct3D □□□□□□

□□□□ DirectDraw

Direct3D Direct3D □□□□¥

DirectDraw Direct3D

Direct3D DirectDraw □□□□ Direct3D

### Direct3Dffff□fff□ffff

Direct3D Direct3D  
 Direct3D 3D DirectDrawSurface  
 DirectDrawSurface Direct3D

```
LPDIRECTDRAW_SURFACE lpDDSsurface;
LPDIRECT3DDEVICE lpD3DDevice;

ddres = lpDD->CreateSurface(&ddsd, &lpDDSsurface,
    NULL);
if (FAILED(ddres))
    .
    .
    .
ddres = lpDDSsurface->QueryInterface(lpGuid,
    &lpD3DDevice);
if (FAILED(ddres))
    .
    .
    .
```

の DirectDrawSurface Direct3D ( Direct3D  
 ) DirectDrawSurface  
 Direct3D  
 DirectDrawSurface ¥ RGB ¥ の

### Direct3Dfffff□fff□ffff

Direct3D DirectDrawSurface の  
 DirectDrawSurface Direct3D

```
LPDIRECTDRAW_SURFACE lpDDSsurface;
LPDIRECT3DTEXTURE lpD3DTexture;

ddres = lpDD->CreateSurface(&ddsd, &lpDDSsurface,
    NULL);
if (FAILED(ddres))
    .
    .
    .
ddres = lpDDSsurface->QueryInterface(
    IID_IDirect3DTexture, &lpD3DTexture);
if (FAILED(ddres))
    .
```

Direct3D  
( Direct3D )  
DirectDrawSurface

## DirectDraw HEL,Direct3D

DirectDraw HEL Z の  
DirectDraw Direct3D の DirectDraw  
Direct3D  
DirectDraw HEL 3D  
DDSCAPS\_3DDEVICE  
3D ¥ の DirectDraw  
3D

**f f X f f f f f f f f f**

X VGA 13 ¥ ¥  
VGA EGA  
256K ¥ ( 13 64K  
) Windows 95 DirectDraw ¥  
X (320×200×8 320×240×8)

X の

X Windows NT

**f f f f f f f f f f**

¥ の ¥ (HEL)  
¥ の

- f f f f f f f f f f f f
- f f f f f f f f f f f f

**f f f f f f f f f f f f**

HEL の red green blue alpha  
の の

f f f f f f f f f f f f “ f f f  
DDPF\_RGB | 1 R: 0x00000000  
DDPF\_PALETTEINDEXED1 G: 0x00000000  
B: 0x00000000

---

		A: 0x00000000
DDPF_RGB	1	R: 0x00000000
DDPF_PALETTEINDEXED1		G: 0x00000000
DDPF_PALETTEINDEXEDTO8		B: 0x00000000
		A: 0x00000000
DDPF_RGB	2	R: 0x00000000
DDPF_PALETTEINDEXED2		G: 0x00000000
		B: 0x00000000
		A: 0x00000000
DDPF_RGB	2	R: 0x00000000
DDPF_PALETTEINDEXED2		G: 0x00000000
DDPF_PALETTEINDEXEDTO8		B: 0x00000000
		A: 0x00000000
DDPF_RGB	4	R: 0x00000000
DDPF_PALETTEINDEXED4		G: 0x00000000
		B: 0x00000000
		A: 0x00000000
DDPF_RGB	4	R: 0x00000000
DDPF_PALETTEINDEXED4		G: 0x00000000
DDPF_PALETTEINDEXEDTO8		B: 0x00000000
		A: 0x00000000
DDPF_RGB	8	R: 0x00000000
DDPF_PALETTEINDEXED8		G: 0x00000000
		B: 0x00000000
		A: 0x00000000
DDPF_RGB	8	R: 0x000000E0
		G: 0x0000001C
		B: 0x00000003
		A: 0x00000000
DDPF_RGB	16	R: 0x000000F0
DDPF_ALPHAPIXELS		G: 0x000000F0
		B: 0x0000000F

---

			A: 0x0000F000
DDPF_RGB	16		R: 0x0000F800 G: 0x000007E0 B: 0x0000001F A: 0x00000000
DDPF_RGB	16		R: 0x0000001F G: 0x000007E0 B: 0x0000F800 A: 0x00000000
DDPF_RGB	16		R: 0x00007C00 G: 0x000003E0 B: 0x0000001F A: 0x00000000
DDPF_RGB   DDPF_ALPHAPIXELS	16		R: 0x00007C00 G: 0x000003E0 B: 0x0000001F A: 0x00008000
DDPF_RGB	24		R: 0x00FF0000 G: 0x0000FF00 B: 0x000000FF A: 0x00000000
DDPF_RGB	24		R: 0x000000FF G: 0x0000FF00 B: 0x00FF0000 A: 0x00000000
DDPF_RGB	32		R: 0x00FF0000 G: 0x0000FF00 B: 0x000000FF A: 0x00000000
DDPF_RGB	32		R: 0x000000FF G: 0x0000FF00 B: 0x00FF0000

		A: 0x00000000
DDPF_RGB	32	R: 0x00FF0000
DDPF_ALPHAPIXELS		G: 0x0000FF00
		B: 0x000000FF
		A: 0xFF000000

DDPF_RGB	32	R: 0x000000FF
DDPF_ALPHAPIXELS		G: 0x0000FF00
		B: 0x00FF0000
		A: 0xFF000000

HEL 3D

のの DirectDraw

### DDSCAPS\_TEXTURE

*ffffff f f f f f f f f f f f f*

DirectX 3 HEL のの red green blue alpha  
のの

<i>ffff ff fff fff</i>	<i>fff “</i>	<i>fff</i>
DDPF_RGB	1	R: 0x00000000
DDPF_PALETTEINDEXED1		G: 0x00000000
		B: 0x00000000
		A: 0x00000000

DDPF_RGB	2	R: 0x00000000
DDPF_PALETTEINDEXED2		G: 0x00000000
		B: 0x00000000
		A: 0x00000000

DDPF_RGB	4	R: 0x00000000
DDPF_PALETTEINDEXED4		G: 0x00000000
		B: 0x00000000
		A: 0x00000000

DDPF_RGB	8	R: 0x00000000
DDPF_PALETTEINDEXED8		G: 0x00000000
		B: 0x00000000
		A: 0x00000000

---

DDPF_RGB	16	R: 0x0000F800 G: 0x000007E0 B: 0x0000001F A: 0x00000000
DDPF_RGB	16	R: 0x00007C00 G: 0x000003E0 B: 0x0000001F A: 0x00000000
DDPF_RGB	24	R: 0x00FF0000 G: 0x0000FF00 B: 0x000000FF A: 0x00000000
DDPF_RGB	24	R: 0x000000FF G: 0x0000FF00 B: 0x00FF0000 A: 0x00000000
DDPF_RGB	32	R: 0x00FF0000 G: 0x0000FF00 B: 0x000000FF A: 0x00000000
DDPF_RGB	32	R: 0x000000FF G: 0x0000FF00 B: 0x00FF0000 A: 0x00000000
HEL 3D	①	Direct3D

## DirectDraw *fff* □ *ffff* Š—

① DirectDraw COM

- *IDirectDraw2 fff ffff*
- *IDirectDrawClipper fff ffff*
- *IDirectDrawPalette fff ffff*
- *IDirectDrawSurface2 fff ffff*

## IDirectDraw2 fff □ ffff

の IDirectDraw2

- DirectDraw ffffff
- IDirectDraw2, %, ,,,,?
- —,,• DirectDraw ffffff
- %o“ fff ff fff “,ff f
- fffff f fff,ff f←fff
- f f• ,”“ffff
- CoCreateInstance,—,,DirectDraw ffffff,

## DirectDraw ffffff

DirectDraw □□□□□□

DirectDraw

DirectDrawSurface DirectDrawPalette DirectDrawClipper

DirectDraw □□□□□□

Windows 95

Windows 95

DirectDraw HAL

Windows 95

GDI の DirectDraw □□□□

□□

(GUID) DirectDraw □□□□□□

GUID

**DirectDrawEnumerate**

DirectDraw □□□□□□

8 /

¥

¥

¥

## IDirectDraw2,%o,□, ,,,,?

DirectX COM

IDirectDraw2

IDirectDraw

IDirectDraw2

IDirectDraw::QueryInterface

```
// IDirectDraw2 fff □ ffff, □□
```

```
LPDIRECTDRAW lpDD;
```

```
LPDIRECTDRAW2 lpDD2;
```

```
ddrval = DirectDrawCreate(NULL, &lpDD, NULL);
```

```
if(ddrval != DD_OK)
```

```
return;
```

```
ddrval = lpDD->SetCooperativeLevel(hwnd,
```

```
DDSCL_NORMAL);
```

```

if(ddsval != DD_OK)
    return;

ddsval = lpDD->QueryInterface(IID_IDirectDraw2,
    (LPVOID *)&lpDD2);
if(ddsval != DD_OK)
    return;

ddscaps.dwCaps = DDSCAPS_OFFSCREENPLAIN;
ddsval = lpDD2->GetAvailableVidMem(&ddscaps, &total,
    &free);
if(ddsval != DD_OK)
    return;

```

**C++のIDirectDraw2**

**IDirectDraw::QueryInterface**

**IDirectDraw2::GetAvailableVidMem**    **IDirectDraw**

**IDirectDraw2::GetAvailableVidMem**    **IDirectDraw2**

**IDirectDrawDraw**

**IDirectDraw2::SetDisplayMode**

**IDirectDraw2::EnumDisplayModes**    **IDirectDraw2**

**IDirectDraw::SetCooperativeLevel**    **CooperativeLevelIDirectDraw**

    の    **IDirectDraw2::SetCooperativeLevel**

**IDirectDraw2::SetDisplayMode** の**IDirectDraw**

**DDSCL\_EXCLUSIVE**

**IDirectDraw::SetCooperativeLevel**

**IDirectDraw::SetDisplayMode**

**DDSCL\_NORMAL**    **IDirectDraw::SetCooperativeLevel**

**IDirectDraw::RestoreDisplayMode**

    DirectDraw

**IDirectDraw2**

∞∞

**IDirectDraw**

**IDirectDraw2**

**IDirectDraw**

**DirectDraw**    **DirectDrawCreate**

DirectDraw □□□□□□

DirectDraw □□□□□□    DirectDraw □□□□□□

    DirectDrawSurface    DirectDrawPalette    DirectDrawClipper の□

□□□□□□    DirectDraw □□□□□□

DirectDraw □□□□□□  
DirectDraw □□□□□□

**DirectDrawCreateClipper** DirectDrawClipper □□  
□□□□ DirectDraw  
DirectDraw □□□□□□

□%o““,fff□ff□,fff□“ff□f

DirectDraw の  
DirectDraw のの 24-bpp 32-bpp の

DirectDraw HEL 24- bpp 32-bpp の  
の

¥  
HEL

Windows DirectDraw

の **IDirectDraw2::SetDisplayMode**  
**IDirectDraw2::EnumDisplayModes**

の

ffff□f□fff¥ff□f←fff

DirectDraw の

DirectDraw GDI の GDI の

DirectDraw GDI GDI の DirectDrawSurface □□□□□□

**IDirectDraw2::GetGDISurface**

GDI DirectDraw ¥ DirectDraw の  
(DDB) DirectDraw の

HAL DIB ¥

f□f•□,““ffff  
**IDirectDraw2::SetDisplayMode**  
¥

の DirectDraw □□□□□□

DirectDrawSurface □□□□□□ ¥  
**IDirectDrawSurface2::Restore**

DirectDraw  
DirectDrawSurface □□□□□□ DirectDraw GDI の

## CoCreateInstance, —, DirectDraw ffffff, □□

**DirectDrawCreate** **CoCreateInstance**  
**IDirectDraw2::Initialize** DirectDraw

```
1 ffff fff,Ž,, NULL,Ž',,CoInitialize,☉, , COM, Š%o,,  
if (FAILED(CoInitialize(NULL)))  
return FALSE;
```

### 2 CoCreateInstance, IDirectDraw2::Initialize

```
ffff,Ž,,DirectDraw ffffff, ,,  
ddrval = CoCreateInstance(&CLSID_DirectDraw,  
NULL, CLSCTX_ALL, &IID_IDirectDraw2, &lpdd);  
if (!FAILED(ddrval))  
ddrval = IDirectDraw2_Initialize(lpdd, NULL);
```

```
,,CoCreateInstance,☉, ,', fff fCLSID_DirectDraw,  
DirectDraw ffffff ffffff fff,fffŽ•Ž,, IID_IDirectDraw2  
fff f, ,,“,DirectDraw fff ffff,Ž•,,  
lpddfff f, Ž“,,,DirectDraw ffffff,Ž, ☉, ,, ☉,, ,,Š , Š%o  
,,,,fffff,•,
```

```
3 DirectDraw ffffff,Ž,‘, IDirectDraw2::Initialize,☉, ,,,,,, ,,ffff,  
DirectDrawCreate Š ,^“,Ž,ffff GUID fff f ,,, NULL ,^ ,Ž,  
DirectDraw ffffff Š%o☉, DirectDrawCreate Š ,Ž,, ,,“,—,Ž—  
%o*,,,,,, DirectDraw ffffff,Š~,ffff,Ž,‘,  
IDirectDraw2::Initialize ffff,☉, ,,,  
DDERR_NOTINITIALIZEDff ,” ,. .
```

**CoUninitialize** COM

CoUninitialize();

## IDirectDrawClipper fff □ ffff

IDirectDrawClipper

**IDirectDrawClipper**

- `fffffff fff`
- `DirectDrawClipperfffffff,←`
- `fff”^DirectDrawClipperfffffff`
- `CoCreateInstance,—,DirectDrawClipperfffffff,`

**fffffff□fff**

DirectDraw DirectDrawClipper

∞∞

DirectDrawClipper

DirectDrawClipper

DirectDraw

DirectDraw HAL

DirectDraw の

HAL

DirectDraw

DirectDraw HAL

のの

¥

¥

の

HAL

のの

¥

DirectDraw ののHAL

**DirectDrawClipperfffffff,←**

DirectDrawClipper □□□□□□

DirectDrawClipper □□□□□□

¥の **DirectDrawSurface2::SetClipper**

DirectDrawClipper □□

□□□□□□

0

DirectDrawClipper □□□□□□

¥

¥

NULL

**IDirectDrawSurface2::SetClipper**

DirectDrawClipper □□□□□□

DirectDrawClipper □□□□□□

---

**Note** “,DirectDrawClipperfffffff “,f fff,‘,  
**IDirectDrawSurface2::SetClipper,%o“’,,E, ,, fffffff,ž ffff,^**  
 “,,ffffff,,~’,,E, ,, fffffff,ž ffff,%o<—,,,,,,

---

**fff”^DirectDrawClipperfffffff**

DirectDraw

DirectDrawClipper

DirectDrawClipper DirectDraw

DirectDrawClipper

**DirectDrawCreateClipper**

DirectDraw

DirectDawClipper

DirectDraw

DirectDrawClipper

DirectDraw

---

**IDirectDraw2::CreateClipper** DirectDrawClipper  
DirectDrawClipper DirectDraw

## CoCreateInstance, — ,,DirectDrawClipper ffffff, □□

DirectDrawClipper □□□□□□ COM の  
**DirectDrawCreateClipper** **IDirectDraw2::CreateClipper**  
**CoGetClassObject**

**CoCreateInstance** **CoCreateInstance**  
DirectDrawClipper □□□□□□

**CoCreateInstance** **IDirectDrawClipper::Initialize**  
DirectDrawClipper

```
ddrval = CoCreateInstance(&CLSID_DirectDrawClipper,  
    NULL, CLSCTX_ALL, &IID_IDirectDrawClipper, &lpClipper);  
if (!FAILED(ddrval))  
    ddrval = IDirectDrawClipper_Initialize(lpClipper,  
        lpDD, 0UL);
```

**CoCreateInstance** *CLSID\_DirectDrawClipper*  
DirectDrawClipper □□□□□□□□□□ *IID\_IDirectDrawClipper*  
*lpClipper*

DirectDrawClipper □□□□□□

の DirectDrawClipper □□□□□□□□□□

**IDirectDrawClipper::Initialize** 0UL *dwFlag*  
0

*lpDD* DirectDrawClipper □□□□□□ DirectDraw  
NULL の DirectDrawClipper □□□□□□

**DirectDrawCreateClipper** DirectDrawClipper

**CoUninitialize** COM

```
CoUninitialize();
```

## IDirectDrawPalette fff□ffff

*IDirectDrawPalette*

- *DirectDrawPaletteffffff*
- ”fffff f fff ,,ffff ’
- ffff,←
- ffff¥fff

## DirectDrawPalette ffffff

DirectDrawPalette 16 256

DirectDrawPalette

DirectDrawSurface

DirectDrawPalette 256  
 0 255 16  
 の  
 の  
 RGB 24 RGB 16  
 16 256

**IDirectDrawPalette::GetEntries**

**IDirectDrawPalette::SetEntries**

の dwFlags

DirectDrawPalette

の

の

**IDirectDrawPalette::SetEntries**

DirectDrawPalette

DirectDrawSurface

DirectDrawPalette

**"ffff□f□fff,,ffff,□"**

¥の  
 DirectDraw

**IDirectDrawSurface2::GetDC**

DirectDrawPalette

Direct3D (

3D

)

¥

**ffff,←**

¥

¥

**IDirectDrawSurface2::SetPalette**

0

¥

¥

NULL

---

**Note** “,ffff “,f fff,‘,  
**IDirectDrawSurface2::SetPalette**,%“,’,,E, ,, ffff,Ž ffff,^“,  
 ,ffffff,, ~’,E, ,, ffff,Ž ffff,%o<—,,,,,

---

**ffff□fff**

DirectDraw 1 (2 ) 2 (4 ) 4 (16 )  
 8 (256 ) ¥

**DDPCAPS\_1BIT**

2  
1

DDPF\_PALETTEINDEXED1

RGB

PALETTEENTRY

2 4

IDirectDraw2::CreatePalette

DDPCAPS\_8BITENTRIES の

DDPCAPS\_4BIT |

DDPCAPS\_8BITENTRIES

PALETTEENTRY の

IDirectDraw2::CreatePalette

LPPALETTEENTRY

**ffff□f□f,,DirectDrawffffŽ—**  
*IDirectDrawPalette*

IDirectDrawPalette

GDI

DirectDraw

の

- (IDirectDraw2::CreatePalette, IDirectDrawPalette::SetEntriesffff, “,,) PALETTEENTRY “, peFlagsfff, “Š,fff,,,,,,,,,
- Windows, “ffff ffff 0,,9 246,,255,ffffff,,),• ,,,,,,,,,,

の Windows の Windows

8

- fffff f f,ffff ffff fff
- fffff f f,,IDirectDraw::CreatePalette,☉, ,
- fffff f f,IDirectDrawPalette::SetEntries,☉, ,

**ffff□f□f,ffff□ffff□fff**

PALETTEENTRY の peFlags

IDirectDraw2::CreatePalette

IDirectDrawPalette::SetEntries

peFlags

3

- Windows “ffff  
~“, Windows, 0,,9 246,,255,,ffff,—  
ffffffff ,,, ffff fff,,ffff, ’,• ,,,,Ž,,,, ,,, ,Š,256 ffff  
ffff,GetSystemPaletteEntries Win32  
Š ,☉, ,, ,,,,,,,,, ffff fff,’ ,,,ffff,Ž ,\*,☉—“,,  
,,,,, ffff f ff,—,,ffff,,•

```

—“,ffff fffffff,fff,,Ž’,, PC_EXPLICIT fff,PALETTEENTRY
“,peFlagsfff ,Š”, ffff,fff,,•—
ffff,ffffff,,peRedfff,fff,, ,,,,ffff fff,,’ Ž ,,,,,, ffff fff,,
< “,Windows,0,255,ffff,,—,,,,,, SetSystemPaletteUse Win32
Š ,œ,,,,,, ,, ,PC_EXPLICIT,‘,PALETTEENTRY
“,0,255,ffff,,Ž’,,,,,,

```

- fff fff ffff  
 ffff fff, ffff fff fffœ%, ,,,,• ,,,  
 fff fff ffff,, PC\_RESERVED fff,ffff fff,Ž’,,  
 Windows,‘ffff fff,~—ffff,•—  
 ffff,fff,,>Ž’, ,,,, ffff fff,ffff,^,, , ’,,,, ‘ffff fff,, -  
 ,,,,,, ”
- ”fff fff ffff  
 ffff fff,• ,,,,ffff,, ”fff fff ffff, ’,PC\_NOCOLLAPSE fff,-  
 ,,,,, ,fff, ‘Š,“ ,•—ffff ffff,Windows,’Š,,,,,,,

PALETTEENTRY の 3

ffff fff	peFlags’	peRed, peGreen, peBlue’
Windows “ffff(0-9, 246-255 ,,0,255,,ffffff)	PC_EXPLICIT	peRed = index, peGreen = 0, peBlue = 0
fff fff ffff	PC_RESERVED   PC_NOCOLLAPSE	’
”fff fff ffff	PC_NOCOLLAPSE	’

ffffff□f□f,,IDirectDraw2::CreatePalette,œ,□,

DirectDraw

IDirectDraw2::CreatePalette

PALETTEENTRY の 256

```

LPDIRECTDRAW          lpDD; // ,,,,□Š%,,,,,
PALETTEENTRY          pPaletteEntry[256];
int                    index;
HRESULT                ddrval;
LPDIRECTDRAWPALETTE  lpDDPal;

// Windows□“ffff,□’,,
for (index = 0; index < 10 ; index++)
{
    // □□,10,□“ffff
    pPaletteEntry[index].peFlags = PC_EXPLICIT;
    pPaletteEntry[index].peRed = index;
    pPaletteEntry[index].peGreen = 0;
    pPaletteEntry[index].peBlue = 0;

    // □œ,10,□“ffff
    pPaletteEntry[index+246].peFlags = PC_EXPLICIT;
    pPaletteEntry[index+246].peRed = index+246;
    pPaletteEntry[index+246].peGreen = 0;
}

```





## **IDirectDrawSurface2,%0,□,,,,,?**

DirectX COM  
*IDirectDrawSurface2*

**IDirectDrawSurface**  
**IDirectDraw::QueryInterface**

```
LPDIRECTDRAWSURFACE lpSurf;  
LPDIRECTDRAWSURFACE2 lpSurf2;  
  
// ffff,□□.  
memset(&ddsd, 0, sizeof(ddsd));  
ddsd.dwSize = sizeof(ddsd);  
ddsd.dwFlags = DDS_DCAPS | DDS_WIDTH | DDS_HEIGHT;  
ddsd.ddsCaps.dwCaps = DDSCAPS_OFFSCREENPLAIN |  
    DDSCAPS_SYSTEMMEMORY;  
ddsd.dwWidth = 10;  
ddsd.dwHeight = 10;  
  
ddrval = lpDD2->CreateSurface(&ddsd, &lpSurf,  
    NULL);  
if(ddrval != DD_OK)  
    return;  
  
ddrval = lpSurf->QueryInterface(  
    IID_IDirectDrawSurface2, (LPVOID *)&lpSurf2);  
if(ddrval != DD_OK)  
    return;  
  
ddrval = lpSurf2->PageLock(0);  
if(ddrval != DD_OK)  
    return;  
  
ddrval = lpSurf2->PageUnlock(0);  
if(ddrval != DD_OK)  
    return;
```

### **IDirectDrawSurface2**

**IDirectDrawSurface2::GetDDInterface**    **IDirectDrawSurface2::PageLock**  
**IDirectDrawSurface2::PageUnlock**        **IDirectDrawSurface**

**f f f f, □ □**

DirectDrawSurface  
¥

¥の DirectDrawSurface  
**IDirectDraw2::CreateSurface** の

**IDirectDraw2::CreateSurface**  
DDSCAPS □□□の dwCaps

DirectDrawSurface  
DDSCAPS\_FLIP

**IDirectDraw2::CreateSurface**

DirectDrawSurface

*IDirectDrawSurface2::DeleteAttachedSurface*

¥ ¥

の

**ffff 1**

GDI DirectDrawSurface  
**DDSURFACEDESC** の **dwHeight** **dwWidth** のののの  
 の

**DDSURFACEDESC** の (

**ddsd)**

```
DDSURFACEDESC ddsd;
ddsd.dwSize = sizeof(ddsd);

// ,,fff,-E,DirectDraw,',,,
ddsd.dwFlags = DDSD_CAPS;

// ffffff□f□fff,fffff
ddsd.ddsCaps.dwCaps = DDSCAPS_PRIMARYSURFACE;
```

**ffff 2**

の  
 のの **DDSURFACEDESC** の ( **ddsd**  
 below)の

```
DDSURFACEDESC ddsd;
ddsd.dwSize = sizeof(ddsd);

// ,,fff,-E,DirectDraw,',,,
ddsd.dwFlags = DDSD_CAPS | DDSD_HEIGHT | DDSD_WIDTH;

// Š',fffff□f□fff,-<,,□
// fff,100□100ffff,,,□
ddsd.ddsCaps.dwCaps = DDSCAPS_OFFSCREENPLAIN;
dwHeight = 100;
dwWidth = 100;
```

DirectDraw の  
**DDSCAPS** の **dwCaps** の  
**DDSCAPS\_SYSTEMMEMORY**



- **IDirectDrawSurface2::Lock**, **IDirectDrawSurface2::Unlock**  
 fffff,Š,,Ž,, Ą,,,, f fff fff,ffffŽ“^‘, ,,  
**IDirectDrawSurface2::Lock**  
 fffff,WIN16,fff,•Ž,,,, **IDirectDrawSurface2::GetDC**  
 fffff,^–“**IDirectDrawSurface2::Lock**,Ą, , WIN16fff,  
**IDirectDrawSurface2::Lock**, **IDirectDrawSurface2::Unlock**,Š, GDI,USE  
 R,ffff,**IDirectDrawSurface2::GetDC**, **IDirectDrawSurface2::ReleaseDC**  
 ,Š,“,,,, Ą,,
- fffffff¥ffff,~‘,,ff ,,,,  
 fff¥ffff¥ffff,Ž,,ffffff¥f f,,,%‘ffff¥ff f¥ffff,Ž‘,,,,, Windows  
 95,f f ff ff ffff Vflatd.386,Ž,  
 ,,ffff,, ,,,ffffff ffff, DirectDraw,~‘,ff fffff,“,,,,, •’,ffffff  
 ¥ffff,ff ,, ff ,ffffff,,,,, , ffff,^Ž’Ž,‘,,%o” ,,

**ffffff □ f □ fff ,GDI ff □ f¥f □ f**

DirectDraw のの  
 ¥ DirectDraw のの のの

の **IDirectDrawSurface2::Flip**

DirectDrawSurface  
 DirectDrawSurface  
 の¥ DirectDrawSurface  
**IDirectDrawSurface2::Flip**

の¥ **IDirectDrawSurface2::Lock**

**IDirectDrawSurface2::Blt**  
**DDERR\_WASSTILLDRAWING** ¥ののの¥

**DirectDrawSurface2::Flip**

¥の

GDI のGDI のの の

20fps GDI

DirectDraw GDI  
 ¥  
 DirectDraw GDI

¥ GDI のの¥ GDI  
 DirectDraw DirectDraw の GDI  
 ¥の

の  
GDIのGDIの

GDI GDI ¥の GDI

- 1 fff¥ffff,,Ž,ffff f fff, ,,
- 2 Šffff f fff(GDI f ff),Šff¥ffff,ffff“,,
- 3 GDI,‘, ;,, Šff ,%Ž%,,,, lpDDSurfaceTargetOverride  
fff f,NULL,fff,,f fff,fffff,,

GDIののGDIのの

pPrimary->Flip(pMiddle);

**f□fff,□Ž**

¥ の

DirectDrawSurface

DirectDrawSurface

DirectDrawSurface

¥の

**DDERR\_SURFACELOST**

¥の¥ の

**IDirectDrawSurface2::Restore**

DirectDrawSurface

の

**□•Š,ff□fff•Š**

RGB

¥

4

FOURCC

)

**IDirectDrawSurface2::GetPixelFormat**

RGB

DDPF\_FOURCC

**DDPIXELFORMAT □□□の dwFourCC □□□**

FOURCC

YUV

DDPF\_YUV

**dwYUVBitCount dwYBits dwUBits dwVBits dwYUVAlphaBits □□□**

RGB

DDPF\_RGB

**dwRGBBitCount**

**dwrRBits**

**dwGBits**

**dwBBits**

**dwRGBAlphaBits**

RGB

DDPF\_RGB

DDPF\_FOURCC

FOURCC

FOURCC

のののの

**ff ¥f fff**

DirectDraw の ¥

の の

¥

¥

¥

¥

¥の ¥

¥

YUV

YUV

¥

¥

¥の ¥

¥の YUV

FOURCC

YUV

¥

**DDCOLORKEY** □□□. **dwColorSpaceLowValue**

**dwColorSpaceHighValue** □□□□□□□□ 3 □□□

V

U

Y

**IDirectDrawSurface2::SetColorKey**

*dwFlags*

¥の の ¥の

**8**

```
// ffff ffff26, ff ¥f □.
dwColorSpaceLowValue = 26;
dwColorSpaceHighValue = 26;
```

**24**

```
// □255, 128, 128, ff ¥f □
dwColorSpaceLowValue = RGBQUAD(255, 128, 128);
dwColorSpaceHighValue = RGBQUAD(255, 128, 128);
```

**FourCC YUV**

```
// YUVff □ Y, 100, 110, š
// U,,, V, 50, 55, š, "%,,,
dwColorSpaceLowValue = YUVQUAD(100, 50, 50);
dwColorSpaceHighValue = YUVQUAD(110, 55, 55);
```

**f □ f □ ffZf □ f □**

Z □□□□

DirectDraw

¥

¥

¥の の

¥の の

¥

Z

の Z  
 Z 0 Z Z  
 Z 2 Z 1 の Z  
 40 Z

**fffff f f ffff,,,. fffff**

DirectDraw ののの¥

16 のの

DirectDraw 256

8

16 の 4

4

のののの

のの

のののの

のののの¥の4

8

3D

¥ ¥のの ¥

**ffff“ ,ffff f f f f f**

□□□□□□

DMA

¥の DDCAPS □□□

12

**dwSVBCaps**

**dwVSBCaps**

**dwSSBCaps**

**dwSVBCKeyCaps**

**dwVSBCKeyCaps**

**dwSSBCKeyCaps**

**dwSVBFXCaps**

**dwVSBFXCaps**

**dwSSBFXCaps**

**dwSVBRops**

**dwVSBRops**

**dwSSBRops**

SVB

¥

¥

□□□□□□

VS

¥

□□□□□□

SSB

□□□□□□

**dwSVBCaps** □□□ **dwCaps** □□□

□□□□□□

¥

の

**dwSVBCKeyCaps** □□□

**dwCKeyCaps** □□□

**dwSVBFXCaps**

□□□ **dwFXCaps** □□□

**dwSVBRops** □□□

□□□□

□□

dwCaps DDCAPS\_CANBLTSYSTEMMEM  
 のの  
 □□□□□□  
 ¥ ¥  
 DirectDraw  
 IDirectDrawSurface2::PageLock

## DirectDraw *ff* □ *ffff*

のの  
 DirectDraw □□□□□□□□ のの  
 SDK DirectDraw のの  
 DDEX の DirectDraw  
 DirectDraw

- *ff* *ffff* 1: DirectDraw,Š'(DDEX1)
- *ff* *ffff* 2: *ffffff*,*fff* *ffff*,“(DDEX2)
- *ff* *ffff* 3: *ffffff* *f* *f* *fff*,,,*ffff*“(DDEX3)
- *ff* *ffff* 4: *ff* ¥*f* ,*ffffff*¥*fff* *fff*(DDEX4)
- *ff* *ffff* 5: *ffff*,“• (DDEX5)

IDirectDraw IDirectDrawSurface  
 DirectX 3 —  
 IDirectDraw2 IDirectDrawSurface2—  
 の QueryInterface  
 IDirectDraw2 ? IDirectDrawSurface2  
 ? IDirectDraw2 IDirectDrawSurface2

---

**Note** ,,ff *ffff*, DDEX*ffffff*ffC++, ,,  
 C*ffff*,Ž,,, *ffff*,',,,,ffff,“• ,, ,, ,, C*ffff*,',,,, ,,vtable*fff*  
*ffff*¥*fff*,,,,this*fff*,%0,•—  
 ,, ,, ,, •, C,Ž,,COM*ffffff*,*fff* ,Ž ,,

---

## Tutorial 1: DirectDraw,Š'

DirectDraw  
 DirectDraw  
 DirectDrawSurface  
 SDK

- DDEX1 のの
- Step 1: DirectDraw*ffffff* ,

- Step 2: `ffff fff, ,*,,CE'`
- Step 3: `ffffff f f,•`
- Step 4: `ffffff f fff ,`
- Step 5: `f fff ,,ffffff`
- Step 6: `f fff,, ,,,`
- Step 7: `f fff,ffffff`
- Step 8: `DirectDrawffffff,Š,“,,%o•`

## Step 1: `DirectDrawffffff, □ □`

```

DirectDraw                                DDEX1                doInit
                                DirectDrawCreate          DirectDrawCreate  3
                                (GUID)                GUID                NULL
                                DirectDraw              DirectDraw
                                の                        の
○○                               DirectDraw
の
の
の
DirectDraw

```

```

ddrval = DirectDrawCreate(NULL, &lpDD, NULL);
if(ddrval == DD_OK)
{
    // lpDD,-E,DirectDrawffffff,,,
}
else
{
    // DirectDrawffffff,□□,,,,,,
}

```

## Step 2: `ffff □ fff, □, *,,CE'`

```

IDirectDraw::SetCooperativeLevel          dwFlags
DDSCL_EXCLUSIVE  DDSCL_FULLSCREEN

```

```

DDSCL_FULLSCREEN

```

```

のの

```

```

DDEX1                ALT + TAB

```

```

の IDirectDraw::SetCooperativeLevel

```

```

HRESULT                ddrval;
LPDIRECTDRAW lpDD;    // DirectDrawCreate,,,,,,□□,,,,,,

```

```

ddrval = lpDD->SetCooperativeLevel(hwnd, DDSCL_EXCLUSIVE |
DDSCL_FULLSCREEN);

```

```

if(ddrval == DD_OK)
{

```

```

        // "\f,GE,,
    }
    else
    {
        // "\f,z",,
        // ,,fffff,,",,,,
    }

```

**IDirectDraw::SetCooperativeLevel** DD\_OK

**IDirectDraw::SetCooperativeLevel**

Windows (GP) ¥ GDI ¥ (HWND) ¥

Windows

DirectDraw

DirectDraw

∞∞

¥

GDI

ALT+TAB

**Step 3: fffffff f f,•**

の

**IDirectDraw::SetDisplayMode**

640×480×8 bpp

```

HRESULT      ddrval;
LPDIRECTDRAW lpDD; // ,,□,.,.,.,

ddrval = lpDD->SetDisplayMode(640, 480, 8);
if(ddrval == DD_OK)
{
    // fffffff¥f□f,•□,□E,,
}
else
{
    // fffffff□f□f,•□,.,.,.,
    // f□f,ff□f,.,.,.,□\,ffff□fff,
    // "\f□f,.,.,.,
}

```

の640×480×8の

(

**IDirectDraw::SetDisplayMode**

DDERR\_INVALIDMODE

**IDirectDraw::EnumDisplayModes**

)

**Step 4: fffffff f fff,□□**

DDEX1

**IDirectDraw::SetCooperativeLevel** DDSCL\_NORMAL

**IDirectDraw::SetCooperativeLevel**

- *f fff* ←, ' <
- *f fff*

の

の

**DDSURFACEDESC** の

の

```
// fff¥ffff, , , ,ž, ffffff f fff, , , ,  
ddsd.dwSize = sizeof(ddsd);  
ddsd.dwFlags = DDSCL_NORMAL | DDSCL_BACKBUFFERCOUNT;  
ddsd.ddsCaps.dwCaps = DDSCAPS_PRIMARYSURFACE |  
    DDSCAPS_FLIP | DDSCAPS_COMPLEX;
```

```
ddsd.dwBackBufferCount = 1;
```

の

**dwSize** の **DDSURFACEDESC** の

DirectDraw

**dwSize** の

**DDSURFACEDESC** の

**dwFlags** の

**DDSURFACEDESC** の DDEX1 の  
**DDSCAPS** (DDSD\_CAPS)  
(DDSD\_BACKBUFFERCOUNT)

**dwFlags**

**dwCaps** の

**DDSCAPS**  
(DDSCAPS\_PRIMARYSURFACE)  
(DDSCAPS\_FLIP) (DDSCAPS\_COMPLEX)

の

¥

¥

¥

DDEX1 の 1

¥

¥

¥

DirectDraw

¥

( 1MB RAM

¥

) **DDSCAPS** の **dwCaps** の

**DDSCAPS\_SYSTEMMEMORY**

**DDSCAPS\_VIDEOMEMORY**

¥

**DDSCAPS\_VIDEOMEMORY**

**IDirectDraw::CreateSurface**

**DDERR\_OUTOFVIDEOMEMORY**

**f fff**, の

**DDSURFACEDESC** の

**DirectDrawCreate**

DirectDraw

*lpDD*

**DDSURFACEDESC** の

**IDirectDraw::CreateSurface**

```

ddrval = lpDD->CreateSurface(&ddsd, &lpDDSPPrimary, NULL);
if(ddrval == DD_OK)
{
    // □,,f□fff,ž,lpDDSPPrimary
}
else
{
    // f□fff,□□,,,,,
    return FALSE;
}

```

*lpDDSPPrimary*  
**IDirectDraw::CreateSurface**

のの¥      の  
**IDirectDrawSurface::GetAttachedSurface**

```

ddscaps.dwCaps = DDSCAPS_BACKBUFFER;
ddrval = lpDDSPPrimary->GetAttachedSurface(&ddcaps, &lpDDSBBack);
if(ddrval == DD_OK)
{
    // lpDDSBBack,□fff¥ffff,,fff
}
else
{
    return FALSE;
}

```

ののDDSCAPS\_BACKBUFFER  
**IDirectDrawSurface::GetAttachedSurface**    *lpDDSBBack*      ¥

**Step 5: f□fff,,fffff**

Windows GDI      ¥      DDEX1      ¥      ¥      のの

```

if (lpDDSPPrimary->GetDC(&hdc) == DD_OK)
{
    SetBkColor(hdc, RGB(0, 0, 255));
    SetTextColor(hdc, RGB(255, 255, 0));
    TextOut(hdc, 0, 0, szFrontMsg, lstrlen(szFrontMsg));
    lpDDSPPrimary->ReleaseDC(hdc);
}

if (lpDDSBBack->GetDC(&hdc) == DD_OK)
{
    SetBkColor(hdc, RGB(0, 0, 255));
    SetTextColor(hdc, RGB(255, 255, 0));
    TextOut(hdc, 0, 0, szBackMsg, lstrlen(szBackMsg));
    lpDDSBBack->ReleaseDC(hdc);
}

```



**IDirectDrawSurface2::GetDC**

¥ SetBkColor SetTextColor

phase ¥ phase 1  
 phase 0 ¥ ¥ phase 0 phase  
 1 の ¥ phase

**IDirectDrawSurface::ReleaseDC****Step 7: fffffff f fff**

□□□□□□ **IDirectDrawSurface::Flip** ¥

```
while(1)
{
    HRESULT ddrval;
    ddrval = lpDDSPPrimary->Flip(NULL, 0);
    if(ddrval == DD_OK)
    {
        break;
    }
    if(ddrval == DDERR_SURFACELOST)
    {
        ddrval = lpDDSPPrimary->Restore();
        if(ddrval != DD_OK)
        {
            break;
        }
    }
    if(ddrval != DDERR_WASSTILLDRAWING)
    {
        break;
    }
}
```

**lpDDSPPrimary** ¥

**IDirectDrawSurface::Flip**

□□□□□□ DD\_OK while

□□□□□□ DDERR\_SURFACELOST

**IDirectDrawSurface::Restore** の

**IDirectDrawSurface::Flip** while



```

if (lpDDPal == NULL)
    goto error;

ddrval = lpDDSPrimary->SetPalette(lpDDPal);

if(ddrval != DD_OK)
    goto error;

// fffffff,fff□ffff,f□f
ddrval = DDReloadBitmap(lpDDSBack, szBackground);

if(ddrval != DD_OK)
    goto error;

```

### Step 1: *ffff*, □ □

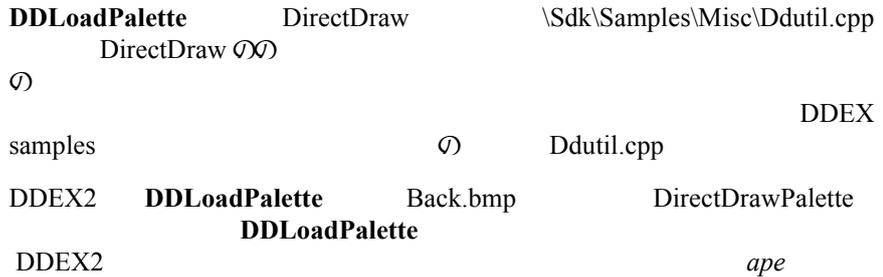
DDEX2 の

```

lpDDPal = DDLoadPalette(lpDD, szBackground);

if (lpDDPal == NULL)
    goto error;

```

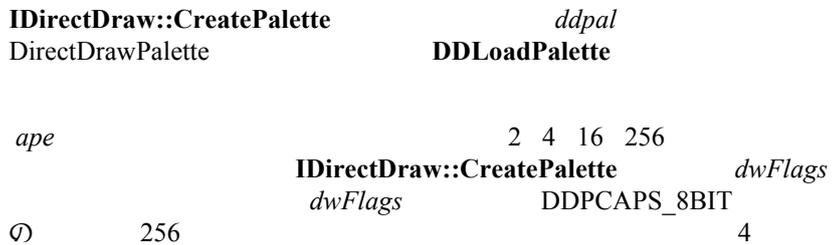


```

DDEX2 の DirectDrawPalette

pdd->CreatePalette(DDPCAPS_8BIT, ape, &ddpal, NULL);
return ddpal;

```



## Step 2: *ffff*, □'

```
                                IDirectDrawSurface::SetPalette
DirectDrawPalette                ddpal   □□□□□□□□□□
の

ddrval = lpDDSPPrimary->SetPalette(lpDDPal);

if(ddrval != DD_OK)
    // SetPaletteŽ”

IDirectDrawSurface::SetPalette                DirectDrawPalette
DirectDrawSurface
```

## Step 3: *fff* □ *ffff*,, *ffffff*, *f* □ *f*

```
DirectDrawPalette                DirectDrawSurface
DDEX2   の                Back.bmp                ¥

// fff □ ffff, ffffff, f □ f
ddrval = DDReLoadBitmap(lpDDSBack, szBackground);

if(ddrval != DD_OK)
    // f □ f, Ž”, ,

DDReLoadBitmap   Ddutil.cpp ののの DirectDraw
(                の
DDLoadBitmap                5:
の                ) DDEX2                szBackground   Back.bmp
    lpDDSBack                ¥                DDReLoadBitmap
                ¥                DDCopyBitmap

DDCopyBitmap   GetObject                ¥の
```

```
// f □ ffff, fff, ,
ddsd.dwSize = sizeof(ddsd);
ddsd.dwFlags = DDSD_HEIGHT | DDSD_WIDTH;
pdds->GetSurfaceDesc (&ddsd);

ddsd   DDSURFACEDESC □□□
DirectDraw   ののの                DDSURFACEDESC □
□□   DDSD_HEIGHT   DDSD_WIDTH                の
IDirectDrawSurface::GetSurfaceDesc
DDEX2                480   640

DDCopyBitmap   ¥

if ((hr = pdds->GetDC (&hdc)) == DD_OK)
```



**dwFlags** □□□ **DDSCAPS** □□□ ‘  
 DDSCAPS\_OFFSCREEN ¥ **DDSCAPS** □□□ ¥ 480 640  
**DDSURFACEDESC** □□□  
**IDirectDraw::CreateSurface**

¥ ¥ ¥  
**IDirectDraw::CreateSurface**  
**DDSCAPS** □□□の **DDSCAPS\_SYSTEMMEMORY**  
**DDSCAPS\_VIDEMEMORY** ¥  
 ¥  
 ¥の¥  
 ¥ **DDSCAPS\_VIDEMEMORY**  
 の **DDERR\_OUTOFVIDEMEMORY**

**Step 2: ffffff□f□f□fff,ffffff,f□f**  
 の DDEX3 Frntback.bmp  
**InitSurfaces** Ddutil.cpp  
**DDCopyBitmap**

```

// ffffff□f□f
hbm = (HBITMAP) LoadImage(GetModuleHandle(NULL), szBitmap,
    IMAGE_BITMAP, 0, 0, LR_CREATEDIBSECTION);

if (hbm == NULL)
    return FALSE;

DDCopyBitmap(lpDDStwo, hbm, 0, 0, 640, 480);
DDCopyBitmap(lpDDStwo, hbm, 0, 480, 640, 480);
DeleteObject(hbm);

return TRUE;

```

Microsoft Paint Frntback.bmp のの  
**DDCopyBitmap** 1 の 1 の(lpDDStwo) 2 の 2 の(lpDDStwo)

**Step 3: fff□ffff,ffffff□f□f□fff,ffff““**  
**WM\_TIMER** DDEX3 ¥  
 rcRect.left = 0;  
 rcRect.top = 0;  
 rcRect.right = 640;  
 rcRect.bottom = 480;



- Step 2:  $\$'$ ,fff fff,

### Step 1: $ff$ $\$f$ $,$ $,$

DirectDraw の doInit の  $\$$   $\$$   $\$$   $\$$  DDEX4 の  $\$$   $\$$   $\$$   $\$$  の

```
//,ffffff,ff$,$f,,
DDSetColorKey(lpDDOne, RGB(0,0,0));

return TRUE;
```

**DDSetColorKey** RGB  $\$$  の  
 RGB (0,0,0) **DDSetColorKey** **DDColorMatch**  
 Ddutil.cpp **DDColorMatch** lpDDOne  
 (0,0)ののの RGB (0,0)  $\$$  の  
 ののの(0,0)  $\$$  の  
**DDSetColorKey** **DDColorMatch**  $\$$   
**DDCOLORKEY**  $\$$ の dwColorSpaceLowValue  $\$$   
 dwColorSpaceHighValue  $\$$   
**IDirectDrawSurface::SetColorKey**  $\$$

**DDSetColorKey** **DDColorMatch** CLR\_INVALID  
 DDEX4 **DDSetColorKey**  $\$$   
 CLR\_INVALID (0,0)  
 $\$$  DDEX4 (0,0)  
 の DDEX4 の (0,0)  
 $\$$  All.bmp  
 (0,0)

```
DDSetColorKey(lpDDOne, CLR_INVALID);
```

DDEX4 の Ddex4.rc の  
 DDEX4 の  $\$$  (0,0)

### Step 2: $\$'$ ,fff $\$$ fff, $\$$ $\$$

DDEX4 の All.bmp  
 updateFrame の 3 3  
 Win32 **GetTickCount**  
**GetTickCount** の

**IDirectDrawSurface::BltFast**  
 (lpDDOne)  $\$$ の  $\$$   $\$$   
 $\$$  DDEX4  $\$$   
 $\$$   $\$$  **IDirectDrawSurface::Flip**

**IDirectDrawSurface::BltFast**

*dwTrans*

DDBLTFAST\_NOCOLORKEY  
 ¥ *dwTrans*  
 DDBLTFAST\_SRCOLORKEY  
 ¥ の *lpDDSDone* □□□□□□

の **updateFrame** の

DDEX4 のの

**Tutorial 5: ffff,““•□**

(DDEX5) 4 (DDEX4) の

DDEX5

- Step 1: ffff ffff,f f
- Step 2: ffff,f f f

**Step 1: ffff□ffff,f□f**

DDEX5 のの All.bmp ののの

```
// ,,,,□,□,□,~ž-,fff,,
for(i=0; i<256; i++)
{
    torusColors[i] = 0;
}

// f□ffff,fff,□E”••,ffff,,□f□ff¥fff□
// E,,,,fffff,~•ž,,
ddsd.dwSize = sizeof(ddsd);
while (lpDDSDone->Lock(NULL, &ddsd, 0, NULL) == DDERR_WASSTILLDRAWING)
;

// f□ff¥fff□f,□□,□ž-□,□f□f,,
for(y=480; y<480+384; y++)
{
    for(x=0; x<640; x++)
    {
        torusColors[((BYTE *)ddsd.lpSurface)[y*ddsd.lPitch+x]] = 1;
    }
}

lpDDSDone->Unlock(NULL);
```

**torusColors** All.bmp のの  
**torusColors** 0

```

torusColors          480  0の¥           のの
DDSURFACEDESC の lpSurface □□□
480  0 (y × IPitch + x)
DDSURFACEDESC          1           y ( row)  IPitch
ののの
torusColors           のの
true                  "*"ddsd.lPitch"
                      (y×IPitch
)

```

## Step 2: *ffff,f□f□f*

DDEX5 updateFrame Tutorial 4 (DDEX4)  
 ¥ 3  
 □□□□□□ updateFrame の doInit

```

// ffff,•□
if(lpDDPal->GetEntries(0, 0, 256, pe) != DD_OK)
{
    return;
}

for(i=1; i<256; i++)
{
    if(!torusColors[i])
    {
        continue;
    }
    pe[i].peRed = (pe[i].peRed+2) % 256;
    pe[i].peGreen = (pe[i].peGreen+1) % 256;
    pe[i].peBlue = (pe[i].peBlue+3) % 256;
}

if(lpDDPal->SetEntries(0, 0, 256, pe) != DD_OK)
{
    return;
}

```

```

IDirectDrawPalette::GetEntries           DirectDrawPalette
pe
DD_OK                                     1
torusColors
DirectDrawPalette
IDirectDrawPalette::SetEntries

```

DDEX5 □□□□□

# DirectDraw

DirectDraw

DirectX SDK

のの

- Stretch  
”””” fffff f fffff fff, ffffff,,“ , ffffff,,“,—  
,, fffff“, ,,—,Ž,
- Donut  
,,”“f f ffff fff,””“f f ffff fff,‘Œ —,Ž,Ž,
- Wormhole  
ffff fff fff,—,Ž,
- Dxview  
fffffŸf ffff,—,’,—,Ž,

DirectDraw

Duel, Iklowns, Foxbear, Palette, Flip2d

## □ “%o, ffffff

SDK DirectDraw ののののの

- fffff,ffff“, ‘Ž“
- ff fff,”,ffff“
- fffffŸf ffff”—,Œ’
- fffffŸfff,,fffff,Š”
- ffff ffffff

## fffff,ffff“, □ ‘Ž“

IDirectDrawSurface2::Flip

□□□□□□□□□□

Ÿ のの

□□□□

□  
DDERR\_WASSTILLDRAWING  
DD\_OK

SDK の IDirectDrawSurface2::Flip

IDirectDrawSurface2::Flip

DDERR\_WASSTILLDRAWING

□□□□□

Ÿ

IDirectDrawSurface2::GetFlipStatus

の□□□□□

DDERR\_WASSTILLDRAWING

□□□□□



**fffffff¥f□ffff”—,OE’**

DirectDraw の DirectDraw ¥ DirectDraw  
 DirectDraw ¥ DirectDraw  
 DirectDraw のののの¥ DirectDraw

¥ **IDirectDraw2::GetCaps**  
 DirectDraw **DDCAPS □□**  
 □の dwCaps □□□ ののの  
**DDCAPS □□□** の  
**DDSCAPS □□□**の DirectDraw のの **DDSCAPS □□□**

**fffffff¥fff,,fffffff,Š”**

¥ ¥  
 ¥のの¥  
 ¥ **DDCAPS □□□**の dwVidMemTotal ¥ののの  
 (の¥ **dwVidMemFree**  
 ) の **IDirectDraw2::GetCaps**  
 ¥ DirectDraw DirectX SDK DirectX  
 ¥ ¥ の ¥  
 のの  
 DirectDraw

**ffff¥fffffff**

ののの¥の¥ ¥

```
// ,,,,fff¥fffffff,fffffff□□fff,□□
ddsd.dwSize = sizeof(ddsd);
ddsd.dwFlags = DDS_DCAPS | DDS_BACKBUFFERCOUNT;
ddsd.ddsCaps.dwCaps = DDSCAPS_PRIMARYSURFACE |
    DDSCAPS_FLIP | DDSCAPS_COMPLEX;
ddsd.dwBackBufferCount = 2;
ddrval = lpDD->CreateSurface(&ddsd, &lpDDSPrimary, NULL);
if(ddrval == DD_OK)
{
    // `^,fff¥fffffff,,fffffff,Š”
    ddscaps.dwCaps = DDSCAPS_BACKBUFFER;
    ddrval = lpDDSPrimary->GetAttachedSurface(&ddscaps,
        &lpDDSBackOne);
    if(ddrval != DD_OK)
        // ,,ff□fff□f,•Ž
        // `^,fff¥fffffff,,fffffff,Š”
    ddscaps.dwCaps = DDSCAPS_BACKBUFFER;
    ddrval = lpDDSPrimary->GetAttachedSurface(&ddscaps,
```

&lpDDBackTwo);

□□□□□□

¥

¥

□□□□□□

¥

**IDirectDrawSurface2::Flip**

DD\_OK

## DirectDrawffffff

Š□

### DirectDrawCreate

```
HRESULT DirectDrawCreate(GUID FAR * lpGUID,  
LPDIRECTDRAW FAR * lpDD, IUnknown FAR * pUnkOuter);
```

DirectDraw

- Ć,, DD\_OK Ź”,,,Ž,ff ,\*,  
**DDERR\_DIRECTDRAWALREADYCREATED**  
**DDERR\_GENERIC**  
**DDERR\_INVALIDDIRECTDRAWGUID**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NODIRECTDRAWHW**  
**DDERR\_OUTOFMEMORY**

*lpGUID*

,,ffff,\*,ff ffff fŽ•Ž (GUID),,ffff  
NULL,,, fffff,ffffff ffff,Ž,

*lpDD*

Ć, ,, Ć,, —Ć,DirectDrawffffff,ffff, Š%o,,ffff,,ffff

*pUnkOuter*

COM %o<”, —,ĆŠ ,,,,, Ć , NULL, ',,,ff ,,

DirectDraw

*lpDD*

**IDirectDraw2::GetCaps**

### DirectDrawCreateClipper

```
HRESULT DirectDrawCreateClipper(DWORD dwFlags,  
LPDIRECTDRAWCLIPPER FAR *lpDDClipper,  
IUnknown FAR *pUnkOuter);
```

DirectDraw

DirectDrawClipper

- DD\_OK DDERR\_INVALIDPARAMS DDERR\_OUTOFMEMORY

*dwFlags*

DirectDrawClipper DirectDraw DirectDrawClipper

*lpDDClipper*

DirectDraw DirectDrawClipper

*pUnkOuter*

DirectDraw DirectDrawClipper

DirectDraw DirectDrawClipper

DirectDrawClipper

DirectDraw DirectDrawClipper

DirectDraw

DirectDrawClipper

**IDirectDraw2::CreateClipper**

**IDirectDraw2::CreateClipper**

## DirectDrawEnumerate

```
HRESULT DirectDrawEnumerate(LPDDENUMCALLBACK lpCallback,
    LPVOID lpContext);
```

DirectDraw GUID  
GDI

- DD\_OK DDERR\_INVALIDPARAMS

*lpCallback*

DirectDraw GUID

*lpContext*

DirectDraw GUID

**f f f f Š**

## Callback

```
BOOL WINAPI lpCallback(GUID FAR * lpGUID,
    LPSTR lpDriverDescription, LPSTR lpDriverName,
    LPVOID lpContext);
```

### DirectDrawEnumerate

- DDENUMRET\_OK DDENUMRET\_CANCEL

*lpGUID*

DirectDraw GUID

---

*lpDriverDescription*  
 ffff,Š,‹ ,Ž,•Ž—,ffff

*lpDriverName*  
 ffff–,Ž,•Ž—,ffff

*lpContext*  
 Š ,Ⓔ, ,,,,,,f ffffŠ ,“,,,ffff fff,’‹,, “,,ffff

## EnumModesCallback

HRESULT WINAPI lpEnumModesCallback(LPDDSURFACEDESC lpDDSurfaceDesc, LPVOID lpContext);

### IDirectDraw2::EnumDisplayModes の

- ‹,’,, ,DDENUMRET\_OK,•, Ž,, ,DDENUMRET\_CANCEL,•,

*lpDDSurfaceDesc*  
 fff ,Ž” , ,“,f f,’‹,,DDSURFACEDESC “,ffff ,,f f,“ , , —”

*lpContext*  
 Š ,Ⓔ, ,,,,,,f ffffŠ ,“,,,ffff fff,’‹,, “,,ffff

## EnumSurfacesCallback

HRESULT WINAPI lpEnumSurfacesCallback(LPDIRECTDRAW\_SURFACE2 lpDDSsurface, LPDDSURFACEDESC lpDDSurfaceDesc, LPVOID lpContext);

### IDirectDrawSurface2::EnumAttachedSurfaces の

- ‹,’,, ,DDENUMRET\_OK,•, Ž,, ,DDENUMRET\_CANCEL,•,

*lpDDSsurface*  
 ,,f fff,ffff,,f fff,ffff

*lpDDSurfaceDesc*  
 ffff,,f fff,‹ ,,DDSURFACEDESC “,ffff

*lpContext*  
 Š ,Ⓔ, ,,,,,,f ffffŠ ,“,,,ffff fff,’‹,, “,,ffff

## fnCallback

HRESULT WINAPI lpfnCallback(LPDIRECTDRAW\_SURFACE2 lpDDSsurface, LPVOID lpContext);

### IDirectDrawSurface2::EnumOverlayZOrders の

- ‹,’,, ,DDENUMRET\_OK,•, Ž,, ,DDENUMRET\_CANCEL,•,

*lpDDSsurface*  
 f fff,f f ff,,,,,f fff,ffff

*lpContext*  
 Š ,Æ, ,,,,,,f ffffŠ ,“,,,,ffff fff,’<, “ ,,,,ffff

## IDirectDraw2

DirectDraw  
**IDirectDraw2**

**IDirectDraw2**  
*DirectDraw*

**IDirectDraw2**

<i>fff,Š,“,</i>	<b>Compact</b> <b>Initialize</b>
<i>ffffff,</i>	<b>CreateClipper</b> <b>CreatePalette</b> <b>CreateSurface</b>
<i>ffff,”—</i>	<b>GetCaps</b>
<i>ffffff¥f f</i>	<b>EnumDisplayModes</b> <b>GetDisplayMode</b> <b>GetMonitorFrequency</b> <b>RestoreDisplayMode</b> <b>SetDisplayMode</b>
<i>ffffff¥ff ff</i>	<b>GetScanLine</b> <b>GetVerticalBlankStatus</b>
<i>,,‘</i>	<b>GetAvailableVidMem</b> <b>GetFourCCCodes</b> <b>WaitForVerticalBlank</b>
<i>“ Š&lt;, ’</i>	<b>SetCooperativeLevel</b>
<i>f fff</i>	<b>DuplicateSurface</b> <b>EnumSurfaces</b> <b>FlipToGDISurface</b> <b>GetGDISurface</b>



DirectDrawClipper                      DirectDrawSurface  
**IDirectDrawSurface2::Blt** **IDirectDrawSurface2::BltBatch**  
**IDirectDrawSurface2::UpdateOverlay**  $\odot$   
DirectDraw                      DirectDrawClipper  
**DirectDrawCreateClipper**  
**IDirectDrawSurface2::GetClipper** **IDirectDrawSurface2::SetClipper**

## IDirectDraw2::CreatePalette

```
HRESULT CreatePalette(DWORD dwFlags,
    LPPALETTEENTRY lpColorTable,
    LPDIRECTDRAWPALETTE FAR * lplpDDPalette,
    IUnknown FAR * pUnkOuter);
```

DirectDrawPalette

- **DD\_OK**    **DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NOCOOPERATIVELEVELSET**  
**DDERR\_NOEXCLUSIVEMODE**  
**DDERR\_OUTOFCAPS**  
**DDERR\_OUTOFMEMORY**  
**DDERR\_UNSUPPORTED**

*dwFlags*

**DDPCAPS\_1BIT**

**DDPCAPS\_2BIT**

**DDPCAPS\_4BIT**

**DDPCAPS\_8BIT**

**DDPCAPS\_8BITENTRIES**

**DDPCAPS\_8BIT**

**DDPCAPS\_ALLOW256**

**DDPCAPS\_1BIT** **DDPCAPS\_2BIT** **DDPCAPS\_4BIT**, **DDPCAPS\_8BIT**, **DDPCAPS\_ALLOW256**

**DDPCAPS\_1BIT** **DDPCAPS\_2BIT** **DDPCAPS\_4BIT**, **DDPCAPS\_8BIT**, **DDPCAPS\_ALLOW256**  
**DDPCAPS\_1BIT** **DDPCAPS\_2BIT** **DDPCAPS\_4BIT**, **DDPCAPS\_8BIT**, **DDPCAPS\_ALLOW256**  
**DDPCAPS\_1BIT** **DDPCAPS\_2BIT** **DDPCAPS\_4BIT**, **DDPCAPS\_8BIT**, **DDPCAPS\_ALLOW256**  
**DDPCAPS\_1BIT** **DDPCAPS\_2BIT** **DDPCAPS\_4BIT**, **DDPCAPS\_8BIT**, **DDPCAPS\_ALLOW256**

**DDPCAPS\_8BIT**

**DDPCAPS\_8BIT** **DDPCAPS\_ALLOW256**

**DDPCAPS\_ALLOW256**

**DDPCAPS\_ALLOW256**, **DDPCAPS\_1BIT**, **DDPCAPS\_2BIT**, **DDPCAPS\_4BIT**, **DDPCAPS\_8BIT**

---

*lpColorTable*

„DirectDrawPaletteffffff, Š%,,,,,2 4 16  
256 „,PALETTEENTRYĀ—,ffff

*lpDDPalette*

„ffff, Ā,, •,, „DirectDrawPaletteffffff,,ffff,ffff

*pUnkOuter*

COM %oĀ, —,ĀŠ ,,,,,, Ā , NULL, ’,,ff ,,

## **IDirectDraw2::CreateSurface**

HRESULT CreateSurface(LPDDSURFACEDESC lpDDSurfaceDesc,  
LPDIRECTDRAW2 FAR \* lpDDSurface,  
IUnknown FAR \* pUnkOuter);

DirectDrawSurface

- Ā,, DD\_OK Ź”,,Ź,ff ,•,  
**DDERR\_INCOMPATIBLEPRIMARY**  
**DDERR\_INVALIDCAPS**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_INVALIDPIXELFORMAT**  
**DDERR\_NOALPHAHW**  
**DDERR\_NOCOOPERATIVELEVELSET**  
**DDERR\_NODIRECTDRAWHW**  
**DDERR\_NOEMULATION**  
**DDERR\_NOEXCLUSIVEMODE**  
**DDERR\_NOFLIPHW**  
**DDERR\_NOMIPMAPHW**  
**DDERR\_NOZBUFFERHW**  
**DDERR\_OUTOFMEMORY**  
**DDERR\_OUTOFVIDEOMEMORY**  
**DDERR\_PRIMARYSURFACEALREADYEXISTS**  
**DDERR\_UNSUPPORTEDMODE**

*lpDDSurfaceDesc*

„,f fff,,—Ā,Ā „,DDSURFACEDESC “,,ffff

*lpDDSurface*

Ā, ,, Ā,, —Ā,DirectDrawSurfacefff,,,, Š%  
„,f fff,,ffff,ffff

*pUnkOuter*

COM %oĀ, —,ĀŠ ,,,,,, Ā , NULL, ’,,ff ,,









*lpDDSurfaceDesc*

*f* *fff*,*Ž*,**DDSURFACEDESC** “*,**fff*”

**IDirectDraw2::RestoreDisplayMode**

**IDirectDraw2::SetDisplayMode**

**IDirectDraw2::RestoreDisplayMode** **IDirectDraw2::EnumDisplayModes**

## **IDirectDraw2::GetFourCCCodes**

**HRESULT** GetFourCCCodes(**LPDWORD** lpNumCodes, **LPDWORD** lpCodes);

DirectDraw FourCC ㊦

- **DD\_OK** *Ž*”*,**Ž*,*ff* ,*\**,  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**

*lpNumCodes*

*lpCodes*,*Ž*”*,*—*,**fff* *fff* , *,**,**,**,**,**,**f* *f*,*Ž*”*,**,* ,  
*lpNumCodes*,—*,**,**,**,**fff*,*,* , *,**,**lpCodes*,”

*lpCodes*

FourCC*f* *f*,”—*,**fff* **NULL**,”*,**,**,*  
*lpNumCodes*,*,* FourCC*f* *f*,*ff* *f*,*,**fff*,*,* *ff* *f*,*,*

## **IDirectDraw2::GetGDISurface**

**HRESULT** GetGDISurface(**LPDIRECTDRAW\_SURFACE FAR \*** lpGDIIDDSurface);

GDI DirectDrawSurface

- **DD\_OK** *Ž*”*,**Ž*,*ff* ,*\**,  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NOTFOUND**

*lpGDIIDDSurface*

GDI,*fffff* *f* *fff*,*ffff* *f*,*,**,**,*DirectDrawSurface*fffff*,*ffff*

**IDirectDraw2::FlipToGDISurface**

## **IDirectDraw2::GetMonitorFrequency**

**HRESULT** GetMonitorFrequency(**LPDWORD** lpdwFrequency);

DirectDraw 100 60Hz 6000

- `DD_OK`, `DDERR_INVALIDOBJECT`, `DDERR_INVALIDPARAMS`, `DDERR_UNSUPPORTED`

*lpdwFrequency*  
 fff Ž” ,Ž,• ,ffff

## IDirectDraw2::GetScanLine

`HRESULT GetScanLine(LPDWORD lpdwScanLine);`

- `DD_OK`, `DDERR_INVALIDOBJECT`, `DDERR_INVALIDPARAMS`, `DDERR_UNSUPPORTED`, `DDERR_VERTICALBLANKINPROGRESS`

*lpdwScanLine*  
 fffffff,CE •%o,,,ffffff,Ž“ ,• ,ffff

**IDirectDraw2::GetVerticalBlankStatus**  
**IDirectDraw2::WaitForVerticalBlank**

## IDirectDraw2::GetVerticalBlankStatus

`HRESULT GetVerticalBlankStatus(LPBOOL lpbIsInVB);`

①

- `DD_OK`, `DDERR_INVALIDOBJECT`, `DDERR_INVALIDPARAMS`

*lpbIsInVB*  
 ’< , ‘,Ž“ ,• ,ffff ,,fff f, ’< ŠŠ,,, , TRUE ,,,, , FALSE  
 ,,,

**IDirectDraw2::WaitForVerticalBlank**

**IDirectDraw2::GetScanLine** **IDirectDraw2::WaitForVerticalBlank**

## IDirectDraw2::Initialize

`HRESULT Initialize(GUID FAR * lpGUID);`

**CoCreateInstance** OLE DirectDraw

- DD\_OK  
**DDERR\_ALREADYINITIALIZED**  
**DDERR\_DIRECTDRAWALREADYCREATED**  
**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NODIRECTDRAWHW**  
**DDERR\_NODIRECTDRAWSUPPORT**  
**DDERR\_OUTOFMEMORY**

*lpGUID*  
 GUID ff ffff ffff ffff  
 COM ¥ ¥  
 DirectDraw DirectDrawCreate  
 DDERR\_ALREADYINITIALIZED IDirectDraw2::Initialize  
 DirectDraw CoCreateInstance  
 DDERR\_NOTINITIALIZED

**CoCreateInstance** **IDirectDraw2::Initialize**  
*CoCreateInstance* *DirectDraw*

**IUnknown::AddRef** **IUnknown::QueryInterface**

**IUnknown::Release**

## **IDirectDraw2::RestoreDisplayMode**

HRESULT RestoreDisplayMode();

¥ **IDirectDraw2::SetDisplayMode**

- DD\_OK  
**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_LOCKEDSURFACES**  
**DDERR\_NOEXCLUSIVEMODE**

**IDirectDraw2::SetDisplayMode**

**IDirectDraw2::EnumDisplayModes** **IDirectDraw2::SetCooperativeLevel**

## IDirectDraw2::SetCooperativeLevel

HRESULT SetCooperativeLevel(HWND hWnd, DWORD dwFlags);

の

- DD\_OK  
DDERR\_EXCLUSIVEMODEALREADYSET  
DDERR\_HWNDALREADYSET  
DDERR\_HWNDSUBCLASSED  
DDERR\_INVALIDOBJECT  
DDERR\_INVALIDPARAMS  
DDERR\_OUTOFMEMORY

*hWnd*

ffff fff,Ž,,fffff¥ffff

*dwFlags*

Ž,fff,’,,, ,^ ,Ž,

DDSCL\_ALLOWMODEX

Mode Xffffff f,Ž—,<%o

DDSCL\_ALLOWREBOOT

”“ fffff f f fŽ, CTRL+ALT+DEL,<’,<%o

DDSCL\_EXCLUSIVE

”“fff,—<

DDSCL\_FULLSCREEN

”“f f, —Ž,‘fffff f fff,’,,%o“, ,,,Ž, GDI,—Ž,,,,,

DDSCL\_NORMAL

ffff fff,’ ,Windowsffff fff,,,’,,,,Ž,

DDSCL\_NOWINDOWCHANGES

fffff fŽ DirectDraw,ffff ffffffff, %o,,,ffff,……,

DDSCL\_EXCLUSIVE

IDirectDraw2::Compact

DDSCL\_EXCLUSIVE

DDSCL\_FULLSCREEN

IDirectDraw2::SetCooperativeLevel

の

DDSCL\_EXCLUSIVE

DDSCL\_NORMAL

DDSCL\_EXCLUSIVE

DDSCL\_FULLSCREEN の

X

DDSCL\_ALLOWMODEX

DDSCL\_FULLSCREEN

DDSCL\_EXCLUSIVE

DDSCL\_ALLOWMODEX

DDSCL\_NORMAL

DDSCL\_ALLOWMODEX

IDirectDraw2::EnumDisplayModes

X

Mode X **IDirectDraw2::SetDisplayMode**  
**IDirectDraw2::SetCooperativeLevel**

Windows X X  
**IDirectDrawSurface2::Lock**  
**IDirectDrawSurface2::Blt** **IDirectDrawSurface2::GetDC**  
DC GDI X  
**IDirectDrawSurface2::GetCaps**  
**IDirectDraw2::EnumDisplayModes**  
DDSURFACEDESC の DDSCAPS の DDSCAPS\_MODEX

**IDirectDraw2::SetCooperativeLevel** DirectDraw  
DDSCL\_NORMAL  
¥ NULL  
Windows DirectDraw

**IDirectDraw::SetDisplayMode** **IDirectDraw::SetCooperativeLevel**  
の *IDirectDraw2* の  
**IDirectDraw**  
**IDirectDraw::RestoreDisplayMode**

1 ”“f f,,,,, DDSCL\_EXCLUSIVEfff,fff,,  
**IDirectDraw::SetCooperativeLevel**,E, ,  
2 fffffff f,• ,,,, **IDirectDraw::SetDisplayMode**,E, ,  
3  
”“f f,%o ,,,,DDSCL\_NORMALfff,<, **IDirectDraw::SetCooperativeLevel**,E, ,

**IDirectDraw2**

**IDirectDraw2::SetDisplayMode** **IDirectDraw2::Compact**  
**IDirectDraw2::EnumDisplayModes**

## **IDirectDraw2::SetDisplayMode**

HRESULT SetDisplayMode(DWORD dwWidth, DWORD dwHeight,  
DWORD dwBPP, DWORD dwRefreshRate, DWORD dwFlags);

¥

- E,,, DD\_OK Ž”,,,Ž,ff ,,  
**DDERR\_GENERIC**  
**DDERR\_INVALIDMODE**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_LOCKEDSURFACES**  
**DDERR\_NOEXCLUSIVEMODE**

---

**DDERR\_SURFACEBUSY**  
**DDERR\_UNSUPPORTED**  
**DDERR\_UNSUPPORTEDMODE**  
**DDERR\_WASSTILLDRAWING**

*dwWidth* and *dwHeight*

,,f f,\*, ,

*dwBPP*

,,f f,fff ffff(bpp)

*dwRefreshRate*

,,f f,fffffff f ,,fff f,,,fff,,, ,,ffff, IDirectDraw  
fff ffff¥f fff,Ž—,,

*dwFlags*

,,fff f,Ⓔ -Ž— 0,fff,,,,,

**IDirectDraw2::SetCooperativeLevel**

DirectDrawSurface

⓪

**IDirectDraw**

*dwRefreshRate*

*dwFlags*

**IDirectDraw::SetDisplayMode** **IDirectDraw::SetCooperativeLevel** ⓪⓪  
*IDirectDraw2*

**IDirectDraw**

**IDirectDraw::RestoreDisplayMode**

1 ""“f f,,,,, DDSCL\_EXCLUSIVEfff,fff,,

**IDirectDraw::SetCooperativeLevel**,Ⓔ, ,

2 fffffff f f,\* ,,, **IDirectDraw::SetDisplayMode**,Ⓔ, ,

3

""“f f,%o ,,,,DDSCL\_NORMALfff,fff,,**IDirectDraw::SetCoopera**  
**tiveLevel** ,Ⓔ, ,

**IDirectDraw2**

⓪

**IDirectDraw2::RestoreDisplayMode**

**IDirectDraw2::GetDisplayMode** **IDirectDraw2::EnumDisplayModes**

**IDirectDraw2::SetCooperativeLevel**

**IDirectDraw2::WaitForVerticalBlank**

HRESULT WaitForVerticalBlank(DWORD dwFlags, HANDLE hEvent);

- Ⓔ,, DD\_OK Ž”,,Ž,ff ,\*,



---

## IDirectDrawClipper::GetClipList

```
HRESULT GetClipList(LPRECT lpRect, LPRGNDATA lpClipList,  
LPDWORD lpdwSize);
```

DirectDrawClipper

- DD\_OK, DDERR\_GENERIC, DDERR\_INVALIDCLIPLIST, DDERR\_INVALIDOBJECT, DDERR\_INVALIDPARAMS, DDERR\_NOCLIPLIST, DDERR\_REGIONTOOSMALL

*lpRect*

fffff fff,ffffff,,,Z-,<E, “,ffff

*lpClipList*

fffff fff,ff E%,Š,RGNDATA “,ffff

*lpdwSize*

fffff fff,fff, ',• ,ffff

**RGNDATA の**

```
typedef struct _RGNDATA {  
    RGNDATAHEADER rdh;  
    char Buffer[1];  
} RGNDATA;
```

**RGNDATA の rdh の RGNDATAHEADER**

```
typedef struct _RGNDATAHEADER {  
    DWORD dwSize;  
    DWORD iType;  
    DWORD nCount;  
    DWORD nRgnSize;  
    RECT rcBound;  
} RGNDATAHEADER;
```

のの

Win32SDK の

**IDirectDrawClipper::SetClipList**

## IDirectDrawClipper::GetHwnd

```
HRESULT GetHwnd(HWND FAR * lphWnd);
```



**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**

*lpbChanged*  
TRUE,fff,,

## **IDirectDrawClipper::SetClipList**

HRESULT SetClipList(LPRGNDATA lpClipList, DWORD dwFlags);

DirectDrawClipper の  
**IDirectDrawSurface2::Blt** **IDirectDrawSurface2::BltBatch**  
**IDirectDrawSurface2::UpdateOverlay**

- DD\_OK  
**DDERR\_CLIPPERISUSINGHWND**  
**DDERR\_INVALIDCLIPLIST**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_OUTOFMEMORY**

*lpClipList*  
LPRGNDATA, NULL, DirectDrawClipper

*dwFlags*  
0, DirectDrawClipper

## **IDirectDrawSurface2::BltFast**

### **RGNDATA の**

```
typedef struct _RGNDATA {  
    RGNDATAHEADER rdh;  
    char Buffer[1];  
} RGNDATA;
```

### **RGNDATA の rdh の RGNDATAHEADER**

```
typedef struct _RGNDATAHEADER {  
    DWORD dwSize;  
    DWORD iType;  
    DWORD nCount;  
    DWORD nRgnSize;  
    RECT rcBound;  
} RGNDATAHEADER;
```

のの Win32 SDK の

**IDirectDrawClipper::GetClipList** **IDirectDrawSurface2::Blt**  
**IDirectDrawSurface2::BltFast** **IDirectDrawSurface2::BltBatch**  
**IDirectDrawSurface2::UpdateOverlay**

## **IDirectDrawClipper::SetHWND**

HRESULT SetHWND(DWORD dwFlags, HWND hWnd);

¥

- DD\_OK DDERR\_INVALIDCLIPLIST  
 DDERR\_INVALIDOBJECT  
 DDERR\_INVALIDPARAMS  
 DDERR\_OUTOFMEMORY

*dwFlags*

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31

*hWnd*

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31

**IDirectDrawClipper::GetHWND**

## **IDirectDrawPalette**

DirectDrawPalette  
**IDirectDrawPalette**  
 ののDirectDrawPalette

**IDirectDrawPalette**

**Initialize**

**GetCaps**

**GetEntries**  
**SetEntries**

**IDirectDrawPalette** COM 3 の  
*IUnknown* **IUnknown**

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



## IDirectDrawPalette::Initialize

```
HRESULT Initialize(LPDIRECTDRAW lpDD, DWORD dwFlags,
    LPPALETTEENTRY lpDDColorTable);
```

DirectDrawPalette

- **DDERR\_ALREADYINITIALIZED**,•

*lpDD*

DirectDraw fffffff, ^-, DirectDraw “, ffff

*dwFlags* and *lpDDColorTable*

,,, fff f, Ā -Ž— 0, fff, ,,,,

COM                      00                      NULL

DirectDrawPalette                      DDERR\_ALREADYINITIALIZED

**IUnknown::AddRef** **IUnknown::QueryInterface**

**IUnknown::Release**

## IDirectDrawPalette::SetEntries

```
HRESULT SetEntries(DWORD dwFlags, DWORD dwStartingEntry,
    DWORD dwCount, LPPALETTEENTRY lpEntries);
```

DirectDrawPalette

- Ā,,, DD\_OK Ž”,,, Ž, ff ,•

**DDERR\_INVALIDOBJECT**

**DDERR\_INVALIDPARAMS**

**DDERR\_NOPALETTEATTACHED**

**DDERR\_NOTPALETTIZED**

**DDERR\_UNSUPPORTED**

*dwFlags*

„ fff f, Ā -Ž— 0, fff, ,,,,

*dwStartingEntry*

fff,,, „, ffff

*dwCount*

• „, ffff ffff

*lpEntries*

ffff ffff, ffff „, ffff ffff, DDPCAPS\_8BITENTRIES

fff, fff, „, ffff ffff, fff, „, „, „, 4fff, „, Šff ff, Ž’, < „, „,

**IDirectDrawPalette::SetEntries**

**IDirectDrawSurface2::SetPalette**

**IDirectDrawPalette::GetEntries** **IDirectDrawSurface2::SetPalette**

---

# IDirectDrawSurface2

## IDirectDrawSurface2

DirectDrawSurface

○○○

○ *DirectDrawSurface*

### IDirectDrawSurface2

*fff,Š,“,*

**Initialize**

**IsLost**

**Restore**

*f fff,ffff*

**AddAttachedSurface**

**DeleteAttachedSurface**

**EnumAttachedSurfaces**

**GetAttachedSurface**

*ffff““*

**Blt**

**BltBatch**

**BltFast**

*ff f*

**GetColorKey**

**SetColorKey**

*ffff ffffff*

**GetDC**

**ReleaseDC**

*f fff ffffff*

**Flip**

*f fff fff*

**Lock**

**PageLock**

**PageUnlock**

**Unlock**

*„‘*

**GetDDInterface**

*f f ff*

**AddOverlayDirtyRect**

**EnumOverlayZOrders**

**GetOverlayPosition**

**SetOverlayPosition**

**UpdateOverlay**

**UpdateOverlayDisplay**

	<b>UpdateOverlayZOrder</b>
‘	<b>GetBltStatus</b> <b>GetFlipStatus</b>
<i>f fff</i> —	<b>GetCaps</b>
<i>f fff ffff</i>	<b>GetClipper</b> <b>SetClipper</b>
<i>f fff</i> ◁	<b>GetPixelFormat</b> <b>GetSurfaceDesc</b>
<i>f fff ffff</i>	<b>GetPalette</b> <b>SetPalette</b>

**IDirectDrawSurface2** COM  
*IUnknown* **IUnknown**  
 ①  
**AddRef**  
**QueryInterface**  
**Release**

## **IDirectDrawSurface2::AddAttachedSurface**

```
HRESULT AddAttachedSurface(
    LPDIRECTDRAW_SURFACE2 lpDDSAttachedSurface);
```

①

- $\mathbb{E},, DD\_OK$  Ž”,,,Ž,ff ,\*,  
**DDERR\_CANNOTATTACHSURFACE**  
**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_SURFACEALREADYATTACHED**  
**DDERR\_SURFACELOST**  
**DDERR\_WASSTILLDRAWING**

*lpDDSAttachedSurface*  
 ffff,, DirectDraw *f fff,,ffff*



**DDERR\_NOBLTHW**  
**DDERR\_NOCLIPLIST**  
**DDERR\_NODDROPSHW**  
**DDERR\_NOMIRRORHW**  
**DDERR\_NORASTEROPHW**  
**DDERR\_NOROTATIONHW**  
**DDERR\_NOSTRETCHHW**  
**DDERR\_NOZBUFFERHW**  
**DDERR\_SURFACEBUSY**  
**DDERR\_SURFACELOST**  
**DDERR\_UNSUPPORTED**

*lpDestRect*

ffff“,,“ ,f fff ,<E, ,,%%%,^,’,<, **RECT** “,ffff

*lpDDSrcSurface*

ffff““ ,“E,, DirectDraw f fff,ffff

*lpSrcRect*

ffff“,,“Ef fff ,<E, ,,%%%, ,’<, **RECT** “,ffff

*dwFlags*

**DDBLT\_ALPHADEST**

ffff ff fff,ffff •, ,,ffff“,ffff ffff,,“ f fff,ffff,,fff  
f ffff f fff,,,,,Ž—,,

**DDBLT\_ALPHADESTCONSTOVERRIDE**

,,ffff““ f fff,ffff ffff,, **DDBLTFX** “,  
**dwAlphaDestConst** fff,Ž—,,

**DDBLT\_ALPHADESTNEG**

ffff’,%%,,“ f fff,“-%%,,,,,•, 0,•“-

**DDBLT\_ALPHADESTSURFACEOVERRIDE**

“ ,ffff ffff,, **DDBLTFX** “,lpDDSAAlphaDest fff,Ž—,,

**DDBLT\_ALPHAEDGEBLEND**

ff f ,%Ž,%‘,fff,ffff ffff,, **DDBLTFX**  
“,dwAlphaEdgeBlend fff,Ž—,,

**DDBLT\_ALPHASRC**

ffff ff fff,ffff •, ,,ffff“,ffff ffff,,“Ef fff,ffff,,fff  
f ffff f fff,,,,,Ž—,,

**DDBLT\_ALPHASRCCONSTOVERRIDE**

“Ef fff,ffff ffff,, **DDBLTFX** “, dwAlphaSrcConst fff,Ž—  
”

**DDBLT\_ALPHASRCNEG**

ffff’,%%,,“Ef fff,“-%%,,,,,•, 0,•“-

**DDBLT\_ALPHASRCSURFACEOVERRIDE**

“E,ffff ffff,, **DDBLTFX** “,lpDDSAAlphaSrc fff,Ž—,,

---

**DDBLT\_ASYNC**

Ž,Ž, ,FIFO ,',,,,ffff“,”“ŠŽ ,, FIFO f ffff,←^,—  
%”,,,,, Ć, ,Ž”,,

**DDBLT\_COLORFILL**

“ f fff ,“ <Ć,—, RGB ,, DDBLTFX “,dwFillColor fff,Ž  
—,,

**DDBLT\_DDFX**

,,ffff“,Ž—,Ć%o,Ž’,,,,, DDBLTFX “,dwDDFX fff,Ž—,,

**DDBLT\_DDROPS**

Win32 API,Š,,,fff — ROPS ,Ž’,,,,,DDBLTFX “,dwDDROPS  
fff,Ž—,,

**DDBLT\_DEPTHFILL**

“ Zffff f fff ,“ <Ć,—, “,,,DDBLTFX “,dwFillDepth  
fff,Ž—,,

**DDBLT\_KEYDEST**

“ f fff,Š~,,,,,ff f ,Ž—,,

**DDBLT\_KEYDESTOVERRIDE**

“ f fff,ff f ,,, DDBLTFX “,dckDestColorkey fff,Ž—,,

**DDBLT\_KEYSRC**

“Ćf fff,Š~,,,,,ff f ,Ž—,,

**DDBLT\_KEYSRCOVERRIDE**

“Ćf fff,ff f ,,, DDBLTFX “,dckSrcColorkey fff,Ž—,,

**DDBLT\_ROP**

,,ffff“,,,ROP , DDBLTFX “, dwROP fff,Ž—,, ,,,, ROP  
, Win32 API ,’<,,,,,, “,,,

**DDBLT\_ROTATIONANGLE**

f fff,%“Š 1/100“^,Ž’,,, ,,,DDBLTFX “, dwRotationAngle  
fff,Ž—,,

**DDBLT\_WAIT**

ffff“,ff ,  
DDERR\_WASSTILLDRAWING•’,,,,•’, , ,ffff“ , ”,,,,,,’,ff ,  
” Ž’ ,,,,—,

**DDBLT\_ZBUFFER**

f fff,“Ć,“ ,,ffff,,Zffff ,,Zffff,fff f,,,DDBLTFX “,  
dwZBufferOpCode fff,Ž—, Zffff ffff“Ž ,,

**DDBLT\_ZBUFFERDESTCONSTOVERRIDE**

“ ,,,, DDBLTFX “,dwZDestConst,, dwZBufferOpCode fff,,,,,  
Zffff,,Zffff fff f,,Ž—, Zffff ffff“Ž ,,

**DDBLT\_ZBUFFERDESTOVERRIDE**

“ ,,,, DDBLTFX “, lpDDSZBufferDest ,,, dwZBufferOpCode  
fff,,,,,Zffff,,Zffff fff f,,Ž—, Zffff ffff“Ž ,,

**DDBLT\_ZBUFFERSRCCONSTOVERRIDE**

“Ć,,,, DDBLTFX “, dwZSrcConst ,,,dwZBufferOpCode

```

fff,,,,Zffff,,Zffff fff f,,Ž—, Zffff ffff“Ž ,
DDBLT_ZBUFFERSRCOVERRIDE
“Ĉ,,, DDBLTFX “, lpDDSZBufferSrc ,, dwZBufferOpCode
fff,,,, Zffff,,Zffff fff f,,Ž—, Zffff ffff“Ž ,

```

```

lpDDBltFx
DDBLTFX “,ffff

```

### **IDirectDrawSurface2::Blt**

DDBLT\_WAIT の

### **IDirectDrawSurface2::BltBatch**

```

HRESULT BltBatch(LPDDBLTBATCH lpDDBltBatch,
    DWORD dwCount, DWORD dwFlags);

```

のの

**IDirectDrawSurface2::Blt**

- Ĉ,, DD\_OK Ž”,,Ž,ff ,;
- DDERR\_GENERIC**
- DDERR\_INVALIDCLIPLIST**
- DDERR\_INVALIDOBJECT**
- DDERR\_INVALIDPARAMS**
- DDERR\_INVALIDRECT**
- DDERR\_NOALPHAHW**
- DDERR\_NOBLTHW**
- DDERR\_NOCLIPLIST**
- DDERR\_NODDROPSHW**
- DDERR\_NOMIRRORHW**
- DDERR\_NORASTEROPHW**
- DDERR\_NOROTATIONHW**
- DDERR\_NOSTRETCHHW**
- DDERR\_NOZBUFFERHW**
- DDERR\_SURFACEBUSY**
- DDERR\_SURFACELOST**
- DDERR\_UNSUPPORTED**

```

lpDDBltBatch
ffff“,,,fff f’,, , DDBLTBATCH “,ffff

```

*dwCount*  
 Ž ,ffff““ ,  
*dwFlags*  
 ,,fff f,Ⓔ -Ž— 0,fff,,,,,

## IDirectDrawSurface2::BltFast

HRESULT BltFast(DWORD dwX, DWORD dwY,  
 LPDIRECTDRAW\_SURFACE2 lpDDSrcSurface, LPRECT lpSrcRect,  
 DWORD dwTrans);

- Ⓔ,, DD\_OK Ž”,,Ž,ff ,\*,  
**DDERR\_EXCEPTION**  
**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_INVALIDRECT**  
**DDERR\_NOBLTHW**  
**DDERR\_SURFACEBUSY**  
**DDERR\_SURFACELOST**  
**DDERR\_UNSUPPORTED**

*dwX* and *dwY*  
 “ f fff ,ffff““,,, •,,,, •

*lpDDSrcSurface*  
 ffff““ ,“Ⓔ DirectDraw f fff,ffff

*lpSrcRect*  
 ffff““,,,“Ⓔf fff ,Ⓔ, ,,%%%, •,’Ⓔ,, **RECT** “,ffff

*dwTrans*  
 “fff

### DDBLTFAST\_DESTCOLORKEY

“ ,ff f ,Ž—,“%Ⓔffff“‘,Ž’,, .

### DDBLTFAST\_NOCOLORKEY

“%o,,,’ ,ff ffff“‘,Ž’,,

### DDBLTFAST\_SRCOLORKEY

“Ⓔ,ff f ,Ž—,“%Ⓔffff“‘,Ž’,,

### DDBLTFAST\_WAIT

ffff“‘,ff ,  
 DDERR\_WASSTILLDRAWING•’,,,,,•’,, ,,,ffff“‘, ”,,,,,,,’ff  
 ,” Ž‘ ,,,-,



**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_SURFACELOST**

*lpContext*

f ffffŠ ,“,, ffff fff,’<, “,,ffff

*lpEnumSurfacesCallback*

ffff,,,,f fff-,Ā, ,, EnumSurfacesCallback Š ,ffff

## **IDirectDrawSurface2::EnumOverlayZOrders**

HRESULT EnumOverlayZOrders(DWORD dwFlags, LPVOID lpContext,  
LPDDENUMSURFACESCALLBACK lpfnCallback);

Ⓞ

- Ā,, DD\_OK Ž”,,Ž,ff ,\*,

**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**

*dwFlags*

^%o,fff,,,,,,,,,,’Ž,

**DDENUMOVERLAYZ\_BACKTOFRONT**

fff,,ffff,Ā,,f f ff,—<,

**DDENUMOVERLAYZ\_FRONTTOBACK**

ffff,,fff,Ā,,f f ff,—<,

*lpContext*

Šf f ff f fff,f ffffŠ ,“,,f f’<ffffff,,ffff

*lpfnCallback*

,,f fff ,f f ff,,f fff-,Ā, ,,, fnCallback Š ,ffff

## **IDirectDrawSurface2::Flip**

HRESULT Flip(  
LPDIRECTDRAW\_SURFACE2 lpDDSurfaceTargetOverride,  
DWORD dwFlags);

**DDSCAPS\_BACKBUFFER**

- Ā,, DD\_OK Ž”,,Ž,ff ,\*,

**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NOFLIPHW**

**DDERR\_NOTFLIPPABLE**  
**DDERR\_SURFACEBUSY**  
**DDERR\_SURFACELOST**  
**DDERR\_UNSUPPORTED**  
**DDERR\_WASSTILLDRAWING**

*lpDDSurfaceTargetOverride*

ffffff,, DirectDraw f fff,ffff  
 ,,fff f,ffffff, NULL,, ,,, **IDirectDrawSurface2::Flip**,ffff',Ⓔ,,ffff,,  
 ,, ,ffff,, ,,fff f,f f fff,,,,,Ž—,,

*dwFlags*

**DDEFLIP\_WAIT**

^”, fffffff f ffff, ‘“ ,,ffffff, ”,,,,  
 DDERR\_WASSTILLDRAWING  
 ff ,,,,\*, fffffff,<,,, ,, ,,fff,fff,,  
 HAL,,DDERR\_WASSTILLDRAWING  
 ff ,Ž,Ž,, **IDirectDrawSurface2::Flip**,ffffff,,,,,‘,,  
 ,, **IDirectDrawSurface2::Flip**,ffffff —, Ⓔ,,  
 DDERR\_SURFACEBUSY,,,,\*,ff ,\*,,,,-,,

DDSCAPS\_FLIP

DDSCAPS\_FRONTBUFFER

⓪

⓪

**IDirectDrawSurface2::Flip**

*lpDDSurfaceTargetOverride*

NULL

**IDirectDrawSurface2::Flip**

**IDirectDrawSurface2::GetFlipStatus**

## **IDirectDrawSurface2::GetAttachedSurface**

HRESULT GetAttachedSurface(LPDDSCAPS lpDDSCaps,  
 LPDIRECTDRAW\_SURFACE2 FAR \* lpDDAttachedSurface);

- Ⓔ,, DD\_OK Ž”,,Ž,ff ,\*,

**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NOTFOUND**  
**DDERR\_SURFACELOST**

*lpDDSCaps*

f fff,f ffff”—,Ž, DDSCAPS “,ffff

---

*lpDDAttachedSurface*

**EnumSurfacesCallback** *f* *ffff*Š , *lpDDSurface*  
*fff* *f*,Ž',,Ⓔ , *DirectDraw* *f* *fff*,*ffff*,, ,,*lpDDSCaps*  
*fff* *f*,Ž',,Ⓔ ,"—,Ž,, *DirectDraw* *f* *fff*,*ffff*,*ffff*  
*DirectDrawSurface*  
*Z* ∅∅

∅

**IDirectDrawSurface2::EnumAttachedSurfaces**

## **IDirectDrawSurface2::GetBltStatus**

**HRESULT** *GetBltStatus*(**DWORD** *dwFlags*);

- *ffff*"*f* *ffff*," ,,, **DD\_OK**  
," *ffff*"*f* *ffff*,*ff* ,, **DDERR\_WASSTILLDRAWING**  
*ffff*"*f* *ffff*," ,,,, **DDERR\_NOBLTHW** Ž",,,Ž,*ff* ," ,  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NOBLTHW**  
**DDERR\_SURFACEBUSY**  
**DDERR\_SURFACELOST**  
**DDERR\_UNSUPPORTED**  
**DDERR\_WASSTILLDRAWING**

*dwFlags*

^%0,*fff*,,,,,,,Ž,

**DDGBS\_CANBLT**

,, *f* *fff*,Š,,*ffff*"",,,Ž ,,,,,,-, ,, *ffff*"",Š—,, **DD\_OK** ," ,

**DDGBS\_ISBLTDONE**

*ffff*"", ,,,,,,-, ,, ,, *f* *fff*, Ⓔ,*ffff*"",Š—,,,,,**DD\_OK** ," ,

## **IDirectDrawSurface2::GetCaps**

**HRESULT** *GetCaps*(**LPDDSCAPS** *lpDDSCaps*);

∅

- Ⓔ,, **DD\_OK** Ž",,,Ž,*ff* ," ,  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**

*lpDDSCaps*  
*f fff,f ffff*—,“, DDSCAPS “,ffff

## IDirectDrawSurface2::GetClipper

HRESULT GetClipper(LPDIRECTDRAWCLIPPER FAR \* lpDDClipper);

DirectDrawClipper の

- DD\_OK Ž”,,Ž,ff ,\*,  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NOCLIPPERATTACHED**

*lpDDClipper*  
*f fff,Š~,,,,, DirectDrawClipper fffffff,,ffff,ffff*  
**IDirectDrawSurface2::SetClipper**

## IDirectDrawSurface2::GetColorKey

HRESULT GetColorKey(DWORD dwFlags, LPDDCOLORKEY lpDDColorKey);

DirectDrawSurface の

- DD\_OK Ž”,,Ž,ff ,\*,  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_NOCOLORKEY**  
**DDERR\_NOCOLORKEYHW**  
**DDERR\_SURFACELOST**  
**DDERR\_UNSUPPORTED**

*dwFlags*  
*,,ff f ,—,Ž’,,*  
**DDCKEY\_DESTBLT**  
*“,ff f ,,,ffff“,,,“ ff f ,,Ž,,,ff ff f,Ž’,,*  
**DDCKEY\_DESTOVERLAY**  
*“,ff f ,,,f f ff —*  
*,,,“ ff f ,,Ž,,,ff ff f,Ž’,,*  
**DDCKEY\_SRCBLT**  
*“,ff f ,,,ffff“,,,“ff f ,,Ž,,,ff ff f,Ž’,,*  
**DDCKEY\_SRCOVERLAY**  
*“,ff f ,,,f f ff —,,,“ff f ,,Ž,,,ff ff f,Ž’,,*

---

*lpDDColorKey*

DirectDrawSurface fffffff,Ž',,,ff f ,E ,',Š",DDCOLORKEY  
“,ffff

**IDirectDrawSurface2::SetColorKey**

## **IDirectDrawSurface2::GetDC**

HRESULT GetDC(HDC FAR \* lphDC);

のの

GDI

- E,,, DD\_OK Ž",,,Ž,ff ,\*,  
**DDERR\_DCALREADYCREATED**  
**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_INVALIDSURFACETYPE**  
**DDERR\_SURFACELOST**  
**DDERR\_UNSUPPORTED**  
**DDERR\_WASSTILLDRAWING**

*lphDC*

ffff fffffff,\*,,ffff,ffff

のの **IDirectDrawSurface2::Lock** の  
**IDirectDrawSurface2::ReleaseDC**

**IDirectDrawSurface2::Lock**

## **IDirectDrawSurface2::GetDDInterface**

HRESULT GetDDInterface(LPVOID FAR \*lpDD);

DirectDraw

- E,,, DD\_OK Ž",,,Ž,ff ,\*,  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**

*lpDD*

DirectDraw ffff,\*,,ffff,ffff E, ,,Ž",,, ,',-E,,,

**IDirectDraw2**

**IDirectDraw**

## IDirectDrawSurface2::GetFlipStatus

HRESULT GetFlipStatus(DWORD dwFlags);

- DD\_OK, DDERR\_WASSTILLDRAWING, DDERR\_INVALIDOBJECT, DDERR\_INVALIDPARAMS, DDERR\_INVALIDSURFACETYPE, DDERR\_SURFACEBUSY, DDERR\_SURFACELOST, DDERR\_UNSUPPORTED

*dwFlags*

DDGFS\_CANFLIP

DDGFS\_ISFLIPDONE

DD\_OK, DDERR\_WASSTILLDRAWING, DDERR\_INVALIDOBJECT, DDERR\_INVALIDPARAMS, DDERR\_INVALIDSURFACETYPE, DDERR\_SURFACEBUSY, DDERR\_SURFACELOST, DDERR\_UNSUPPORTED

DDGFS\_ISFLIPDONE

DD\_OK, DDERR\_WASSTILLDRAWING, DDERR\_INVALIDOBJECT, DDERR\_INVALIDPARAMS, DDERR\_INVALIDSURFACETYPE, DDERR\_SURFACEBUSY, DDERR\_SURFACELOST, DDERR\_UNSUPPORTED

IDirectDrawSurface2::Flip

## IDirectDrawSurface2::GetOverlayPosition

HRESULT GetOverlayPosition(LPLONG lpIX, LPLONG lpIY);

DDSCAPS\_OVERLAY

DDSCAPS\_OVERLAY

DDSCAPS\_OVERLAY

- DDERR\_GENERIC, DDERR\_INVALIDOBJECT, DDERR\_INVALIDPARAMS, DDERR\_INVALIDPOSITION, DDERR\_NOOVERLAYDEST, DDERR\_NOTAOVERLAYSURFACE, DDERR\_OVERLAYNOTVISIBLE, DDERR\_SURFACELOST

*lpIX and lpIY*

DD\_OK, DDERR\_GENERIC, DDERR\_INVALIDOBJECT, DDERR\_INVALIDPARAMS, DDERR\_INVALIDPOSITION, DDERR\_NOOVERLAYDEST, DDERR\_NOTAOVERLAYSURFACE, DDERR\_OVERLAYNOTVISIBLE, DDERR\_SURFACELOST

IDirectDrawSurface2::SetOverlayPosition

IDirectDrawSurface2::UpdateOverlay



```

lpDDSurfaceDesc
    f fff, 'Ž', DDSURFACEDESC 'fff
    DDSURFACEDESC

```

## IDirectDrawSurface2::Initialize

```

HRESULT Initialize(LPDIRECTDRAW lpDD,
    LPDDSURFACEDESC lpDDSurfaceDesc);

```

DirectDrawSurface

- **DDERR\_ALREADYINITIALIZED**,

```

lpDD
    DirectDraw ffffff, DirectDraw 'fff
lpDDSurfaceDesc
    f fff,,,, ' , DDSURFACEDESC 'fff
    COM
    DirectDrawSurface
    DDERR_ALREADYINITIALIZED

```

**IUnknown::AddRef** **IUnknown::QueryInterface**  
**IUnknown::Release**

## IDirectDrawSurface2::IsLost

```

HRESULT IsLost();

```

DirectDrawSurface

- **DD\_OK**, **DDERR\_INVALIDOBJECT**,  
**DDERR\_INVALIDPARAMS**,  
**DDERR\_SURFACELOST**

```

DirectDrawSurface
DDERR_SURFACELOST

```

## IDirectDrawSurface2::Restore

## IDirectDrawSurface2::Lock

```

HRESULT Lock(LPRECT lpDestRect, LPDDSURFACEDESC lpDDSurfaceDesc,
    DWORD dwFlags, HANDLE hEvent);

```

①

- DD\_OK Ž”,,Ž,ff ,•  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_OUTOFMEMORY**  
**DDERR\_SURFACEBUSY**  
**DDERR\_SURFACELOST**  
**DDERR\_WASSTILLDRAWING**

*lpDestRect*

fff,,f fff—^,Ž, RECT “,ffff

*lpDDSurfaceDesc*

f fff,,,, •,Ž’,, DDSURFACEDESC “,ffff

*dwFlags*

**DDLOCK\_EVENT**

**IDirectDrawSurface2::Lock**,—<,,f fff fff,ffff,•,,,%0”,,,,ffff,<  
“,,fff ffff ffff,**IDirectDrawSurface2::Lock** “,“,,, fff,,  
,,fff,fff,f fff ,• ,,,, ffff,FIFO ,^<,,,

**DDLOCK\_READONLY**

fff,,f fff,“ , —,,,,,Ž,fff

**DDLOCK\_SURFACEMEMORYPTR**

Ž’,,<E, “,—E,fff ffff,•,,,,,,•,fff  
<E,Ž’,,, ~” ,f fff,,ffff,•,,, fffff,,,,fff,Ž’,,

**DDLOCK\_WAIT**

’ ffff“ —,ff ‘,,,fff,“,,, ffff,’,, —,  
DDERR\_WASSTILLDRAWING ff ,•,  
,,,,fff,fff,, fff,“,,, DDERR\_SURFACEBUSY,, •,ff ,” ,,,,  
**IDirectDrawSurface2::Lock**, Ž’,,

**DDLOCK\_WRITEONLY**

fff,,f fff, , —,,,,,Ž,

*hEvent*

f fff,fff, ”,,, Š,fff,,ffff ffff,ffff

**IDirectDrawSurface2::Unlock**

**IDirectDrawSurface2::Unlock**

②

DDERR\_SURFACEBUSY

DDERR\_LOCKEDSURFACES







ののの

DirectDrawClipper

DirectDrawClipper

**IDirectDrawSurface2::Blt**

**IDirectDrawSurface2::BltBatch**

**IDirectDrawSurface2::UpdateOverlay** のの

DirectDrawSurface

DirectDrawClipper

DirectDrawClipper

の

**IDirectDrawSurface2::GetClipper**

## **IDirectDrawSurface2::SetColorKey**

`HRESULT SetColorKey(DWORD dwFlags, LPDDCOLORKEY lpDDColorKey);`

DirectDrawSurface の

- `DD_OK` `DDERR_GENERIC` `DDERR_INVALIDOBJECT` `DDERR_INVALIDPARAMS` `DDERR_INVALIDSURFACETYPE` `DDERR_NOOVERLAYHW` `DDERR_NOTAOVERLAYSURFACE` `DDERR_SURFACELOST` `DDERR_UNSUPPORTED` `DDERR_WASSTILLDRAWING`

**DDERR\_GENERIC**

**DDERR\_INVALIDOBJECT**

**DDERR\_INVALIDPARAMS**

**DDERR\_INVALIDSURFACETYPE**

**DDERR\_NOOVERLAYHW**

**DDERR\_NOTAOVERLAYSURFACE**

**DDERR\_SURFACELOST**

**DDERR\_UNSUPPORTED**

**DDERR\_WASSTILLDRAWING**

*dwFlags*

`DDCKEY_COLORSPACE`

**DDCKEY\_COLORSPACE**

`DDCKEY_DESTBLT`

**DDCKEY\_DESTBLT**

`DDCKEY_DESTOVERLAY`

**DDCKEY\_DESTOVERLAY**

`DDCKEY_SRCBLT`

**DDCKEY\_SRCBLT**

`DDCKEY_SRCOVERLAY`

**DDCKEY\_SRCOVERLAY**

```

f f ff —,““ff f ,,Ž—
,,ff f ,,““ ff ff f, “,Ž’,,,, fff,,

```

*lpDDColorKey*

```

DirectDrawSurface fffffff, ,,ff f ’,Š,,, DDCOLORKEY
“,ffff

```

**IDirectDrawSurface2::GetColorKey**

## **IDirectDrawSurface2::SetOverlayPosition**

```

HRESULT SetOverlayPosition(LONG lX, LONG lY);

```

⓪

- Ⓔ,,, DD\_OK Ž”,,,Ž,ff ,•,  
**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_SURFACELOST**  
**DDERR\_UNSUPPORTED**

*lX* and *lY*

```

,,,ffffff •,,,ffffff •

```

**IDirectDrawSurface2::GetOverlayPosition**

**IDirectDrawSurface2::UpdateOverlay**

## **IDirectDrawSurface2::SetPalette**

```

HRESULT SetPalette(LPDDIRECTDRAWPALETTE lpDDPalette);

```

DirectDrawPalette

⓪⓪

- Ⓔ,,, DD\_OK Ž”,,,Ž,ff ,•,  
**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_INVALIDSURFACETYPE**  
**DDERR\_NOEXCLUSIVEMODE**  
**DDERR\_NOPALETTEATTACHED**  
**DDERR\_NOPALETTEHW**  
**DDERR\_NOT8BITCOLOR**  
**DDERR\_SURFACELOST**  
**DDERR\_UNSUPPORTED**

---

*lpDDPalette*  
DirectDrawPalette “,ffff  
ののの

**IDirectDrawSurface2::GetPalette** **IDirectDraw2::CreatePalette**

## **IDirectDrawSurface2::Unlock**

HRESULT Unlock(LPVOID lpSurfaceData);

DirectDraw

- $\mathbb{C}$ ,, DD\_OK Ž”,,,Ž,ff ,\*,  
**DDERR\_GENERIC**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_INVALIDRECT**  
**DDERR\_NOTLOCKED**  
**DDERR\_SURFACELOST**

*lpSurfaceData*

**IDirectDrawSurface2::Lock**ffff,,,\*,,ffff,ffff “,f fff,^,“ <mathbb{C},,,\*  
, **IDirectDrawSurface2::Lock**( $\mathbb{C}$ , ,,%”,,, ,ffff, **IDirectDrawSurface**  
**2::Lock**,, **IDirectDrawSurface2::Unlock**( $\mathbb{C}$ , ,, $\mathbb{C}$ ,\*,  
**IDirectDrawSurface2::Lock**

## **IDirectDrawSurface2::UpdateOverlay**

HRESULT UpdateOverlay(LPRECT lpSrcRect,  
LPDIRECTDRAWSURFACE2 lpDDDestSurface,  
LPRECT lpDestRect, DWORD dwFlags,  
LPDDOVERLAYFX lpDDOverlayFx);

の の  
DDSCAPS\_OVERLAY

- $\mathbb{C}$ ,, DD\_OK Ž”,,,Ž,ff ,\*,  
**DDERR\_GENERIC**  
**DDERR\_HEIGHTALIGN**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_INVALIDRECT**  
**DDERR\_INVALIDSURFACETYPE**  
**DDERR\_NOSTRETCHHW**  
**DDERR\_NOTAOVERLAYSURFACE**



```

    “ f fff,☉,•,•,•,ff f ,Ž—,,
DDOVER_KEYDESTOVERRIDE
    “ f fff,ff f ,, DDOVLAYFX “,dckDestColorkey fff,Ž—,,
DDOVER_KEYSRC
    “☉f fff,☉,•,•,•,ff f ,Ž—,,
DDOVER_KEYSRCOVERRIDE
    “☉f fff,ff f ,, DDOVLAYFX “,dckSrcColorkey fff,Ž—,,
DDOVER_SHOW
    ,,f f ff,ff,,
DDOVER_ZORDER
    ,,f f ff,Zf f ,, DDOVLAYFX “,dwZOrderFlags fff,Ž—
    ,, lpDDSRelative fff, dwZOrderFlags
    fff,DDOVERZ_INSERTINBACKOF
    ,,,DDOVERZ_INSERTINFRONTOF ,fff,•,•,•,Ž—,,

```

```

lpDDOverlayFx
DDOVLAYFX “,Ž ,•,•,•,

```

## IDirectDrawSurface2::UpdateOverlayDisplay

```

HRESULT UpdateOverlayDisplay(DWORD dwFlags);

```

```

○○○
○

```

- ☉,•,•, DD\_OK Ž”•,•,Ž,ff ,•,
  - DDERR\_INVALIDOBJECT**
  - DDERR\_INVALIDPARAMS**
  - DDERR\_INVALIDSURFACETYPE**
  - DDERR\_UNSUPPORTED**

```

dwFlags
—,•,•, fff ^%0,fff,•,•,•,•,•,Ž,
DDOVER_REFRESHDIRTYRECTS
,,“ ,•,•,•,•,•,^,, ,•,•,☉,fff,Ž—
,, ffffff, ,•,•,•,☉fff,fff,,
DDOVER_REFRESHALL
,,☉,fff,~Ž, f f ff ffffff,Š‘, ,•,•,•,☉fff,fff,,

```

**IDirectDrawSurface2::AddOverlayDirtyRect**

## IDirectDrawSurface2::UpdateOverlayZOrder

```
HRESULT UpdateOverlayZOrder(DWORD dwFlags,
    LPDIRECTDRAWSURFACE2 lpDDSReference);
```

Z

- DD\_OK, DDERR\_INVALIDOBJECT, DDERR\_INVALIDPARAMS, DDERR\_NOTAOVERLAYSURFACE

*dwFlags*

^%0,fff,,,,,,',Ž,

**DDOVERZ\_INSERTINBACKOF**

„f f ff,f f ff ff f,Ž f f ff,„,,

**DDOVERZ\_INSERTINFRONTOF**

„f f ff,f f ff ff f,Ž f f ff,‘,“,,

**DDOVERZ\_MOVEBACKWARD**

„f f ff,f f ff ff f,,,„•,“,,

**DDOVERZ\_MOVEFORWARD**

„f f ff,f f ff ff f,,,‘,“,,

**DDOVERZ\_SENDBACK**

„f f ff,f f ff ff f,„•,“,,

**DDOVERZ\_SENDFRONT**

„f f ff,f f ff ff f,‘,“,,

*lpDDSReference*

f f ff ff f,,,“,^,Ž—,,DirectDraw f fff,ffff  
 ,,fff f, DDOVERZ\_INSERTINBACKOF,,, DDOVERZ\_INSERTINFRO  
 NTOF ,,,,•—,,

**IDirectDrawSurface2::EnumOverlayZOrders**

□“

## DDBLTBATCH

```
typedef struct _DDBLTBATCH{
    LPRECT          lprDest;
    LPDIRECTDRAWSURFACE lpDDSrc;
    LPRECT          lprSrc;
    DWORD           dwFlags;
    LPDDBLTFX       lpDDBltFx;
} DDBLTBATCH, FAR *LPDDBLTBATCH;
```

**IDirectDrawSurface2::BltBatch**

---

**lprDest**

ffff“ ,’<, RECT “,ffff

**lpDDSrc**

ffff“Ĉ,’,, DirectDrawSurfacefffff,ffff

**lprSrc**

ffff“Ĉ,Ĉ,’,, RECT “,ffff

**dwFlags**

ffff Ĉfff

**DDBLT\_ALPHADEST**

„ffff“,,,ffff“ f fff,ffff ffff,, ffff ff fff,,ffff •,,,f  
fff ffff f fff,,,,,Ž—,,

**DDBLT\_ALPHADESTCONSTOVERRIDE**

„ffff“,,,ffff“ f fff,ffff ffff,, DDBLTFX “,  
dwAlphaDestConst fff,Ž—,,

**DDBLT\_ALPHADESTNEG**

ffff’,%o,,“ f fff,“-%o,,,,,•, 0,“-

**DDBLT\_ALPHADESTSURFACEOVERRIDE**

„ffff“,,,ffff“ f fff,ffff ffff,, DDBLTFX “,  
lpDDSAlphaDest fff,Ž—,,

**DDBLT\_ALPHAEDGEBLEND**

ff f ,<, ,%o’,fff,ffff ffff,,DDBLTFX “,  
dwAlphaEdgeBlend fff,Ž—,,

**DDBLT\_ALPHASRC**

„ffff“,,,ffff“ f fff,ffff ffff,, ffff ff fff,,ffff •,,,f  
fff ffff f fff,,,,,Ž—,,

**DDBLT\_ALPHASRCCONSTOVERRIDE**

„ffff“,,,ffff“Ĉf fff,ffff ffff,, DDBLTFX “,  
dwAlphaSrcConst fff,Ž—,,

**DDBLT\_ALPHASRCNEG**

ffff’,%o,,“Ĉf fff,“-%o,,,,,•, 0,“-

**DDBLT\_ALPHASRCSURFACEOVERRIDE**

„ffff“,,,ffff“Ĉf fff,ffff ffff,, DDBLTFX “,  
lpDDSAlphaSrc fff,Ž—,,

**DDBLT\_ASYNC**

FIFO f ffff’,Ž , , ,ffff“,”Š,Ž , , ,FIFO f ffff,←  
^,,,, Ĉ, ,Ž”,,

**DDBLT\_COLORFILL**

ffff“ f fff ,ffff“ ĈĈ,-, RGB ,,, DDBLTFX “,  
dwFillColor fff,Ž—,,

**DDBLT\_DDFX**

„ffff“Ž—,,Ĉ%o,Ž’,, DDBLTFX “, dwDDFX fff,Ž—,,

**DDBLT\_DDROPS**

Win32 API ,^•,,fff — ROPs ,Ž’,,,,,DDBLTFX “,



```

        DWORD dwRotationAngle;
        DWORD dwZBufferOpCode;
        DWORD dwZBufferLow;
        DWORD dwZBufferHigh;
        DWORD dwZBufferBaseDest;
        DWORD dwZDestConstBitDepth;
union
{
        DWORD dwZDestConst;
        LPDIRECTDRAWSURFACE lpDDSZBufferDest;
};
        DWORD dwZSrcConstBitDepth;
union
{
        DWORD dwZSrcConst;
        LPDIRECTDRAWSURFACE lpDDSZBufferSrc;
};
        DWORD dwAlphaEdgeBlendBitDepth;
        DWORD dwAlphaEdgeBlend;
        DWORD dwReserved;
        DWORD dwAlphaDestConstBitDepth;
union
{
        DWORD dwAlphaDestConst;
        LPDIRECTDRAWSURFACE lpDDSAlphaDest;
};
        DWORD dwAlphaSrcConstBitDepth;
union
{
        DWORD dwAlphaSrcConst;
        LPDIRECTDRAWSURFACE lpDDSAlphaSrc;
};
union
{
        DWORD dwFillColor;
        DWORD dwFillDepth;
        LPDIRECTDRAWSURFACE lpDDSPattern;
};
        DDCOLORKEY ddckDestColorkey;
        DDCOLORKEY ddckSrcColorkey;
} DDBLTFX, FAR* LPDDBLTFX;

```

## IDirectDrawSurface2::Blt

### IDirectDrawSurface2::BltBatch

#### DDBLTBATCH の

##### dwSize

“‘,, ,fff, “Ž,,‘ ’,,,,,,,,,,,,,”

##### dwDDFX

Ⓔ%o —,fff,Ž’,fff

##### DDBLTFX\_ARITHSTRETCHY

,ffff“,,,,Ž,%o,%oŽ ’,Ž—,,

**DDBLTFX\_MIRRORLEFTRIGHT**

Ž ,f fff,%o“, , ,ffff“ ,f fff, ,%,“Ž,,

**DDBLTFX\_MIRRORUPDOWN**

Ž ,f fff,%o“, , ,ffff“ ,f fff, ,%,“Ž,,

**DDBLTFX\_NOTEARING**

,ffff“ ,,,,fffff,”,,,, —,,

**DDBLTFX\_ROTATE180**

,ffff“ ,Š,f fff,ŽĚ%o,,180“%o“,,,

**DDBLTFX\_ROTATE270**

,ffff“ ,Š,f fff,ŽĚ%o,,270“%o“,,,

**DDBLTFX\_ROTATE90**

,ffff“ ,Š,f fff,ŽĚ%o,,90“%o“,,,

**DDBLTFX\_ZBUFFERBASEDEST**

,ffff“ ,Š, ”Š,,‘,ffff“Ě,’,,,ffff“ ,ffff“ ,’,,,,,,dwZBufferBaseDest fff,%o,,

**DDBLTFX\_ZBUFFERRANGE**

,ffff“ ,Š,ffff“Ěf fff,,\* ,fff,ĚŠ,Ž’,,”^,,,dwZBufferLow,,, dwZBufferHigh fff,Ž—,,

**dwROP**

Win32 fff —

**dwDDROP**

DirectDraw fff —

**dwRotationAngle**

ffff“ ,%o“Š“

**dwZBufferOpCode**

Zffff”Š

**dwZBufferLow**

Zffff, ’ĚŠ

**dwZBufferHigh**

Zffff, ĚŠ

**dwZBufferBaseDest**

Zffff,ffff“ Š”

**dwZDestConstBitDepth**

ffff“ ,’ ,fff “

**dwZDestConst**

ffff“ Zffff,,Ž—,,,’

**lpDDSZBufferDest**

ffff“ Zffff,,Ž—,,f fff

**dwZSrcConstBitDepth**

ffff“Ě,’ ,fff “

**dwZSrcConst**

ffff“ĚZffff,,Ž—,,,’

---

**lpDDSZBufferSrc**

ffff““EZffff,,Ž—,,f fff

**dwAlphaEdgeBlendBitDepth**

ffff fff ,,,fff “

**dwAlphaEdgeBlend**

fff ,,,Ž,,ffff’

**dwReserved**

—,,,—,,,

**dwAlphaDestConstBitDepth**

ffff““ ffff’ ,fff “

**dwAlphaDestConst**

ffff““ ffff ffff,,Ž,,’

**lpDDSAlphaDest**

ffff““ ffff ffff,,Ž—,,f fff

**dwAlphaSrcConstBitDepth**

ffff““Effff’ ,fff “

**dwAlphaSrcConst**

ffff““Effff ffff,,Ž,,’

**lpDDSAlphaSrc**

ffff““Effff ffff,,Ž—,,f fff

**dwFillColor**

DDBLT\_COLORFILL ,Ž’,,Ž f fff,• ,,,Ž—  
,, ,Ž’ ,’, f fff fff,^,,RGB, ’,,ffff ffffff,,

**dwFillDepth**

Zffff, “

**lpDDSPattern**

ff f,,Ž—,,f fff ff f,ffff““E,ffff““ ,ffff,,“ ,ffff““ ,Ž—,,

**ddckDestColorkey**

— ,ffff““ ff f

**ddckSrcColorkey**

— ,ffff““Efff f

**DDCAPS**

```
typedef struct _DDCAPS{
    DWORD    dwSize;
    DWORD    dwCaps;
    DWORD    dwCaps2;
    DWORD    dwCKKeyCaps;
    DWORD    dwFXCaps;
    DWORD    dwFXAlphaCaps;
    DWORD    dwPalCaps;
    DWORD    dwSVCaps;
    DWORD    dwAlphaBltConstBitDepths;
    DWORD    dwAlphaBltPixelBitDepths;
    DWORD    dwAlphaBltSurfaceBitDepths;
    DWORD    dwAlphaOverlayConstBitDepths;
```

```

DWORD    dwAlphaOverlayPixelBitDepths;
DWORD    dwAlphaOverlaySurfaceBitDepths;
DWORD    dwZBufferBitDepths;

DWORD    dwVidMemTotal;
DWORD    dwVidMemFree;
DWORD    dwMaxVisibleOverlays;
DWORD    dwCurrVisibleOverlays;
DWORD    dwNumFourCCCodes;
DWORD    dwAlignBoundarySrc;
DWORD    dwAlignSizeSrc;
DWORD    dwAlignBoundaryDest;
DWORD    dwAlignSizeDest;
DWORD    dwAlignStrideAlign;
DWORD    dwRops[DD_ROP_SPACE];
DDSCAPS  ddsCaps;
DWORD    dwMinOverlayStretch;
DWORD    dwMaxOverlayStretch;
DWORD    dwMinLiveVideoStretch;

DWORD    dwMaxLiveVideoStretch;
DWORD    dwMinHwCodecStretch;
DWORD    dwMaxHwCodecStretch;
DWORD    dwReserved1;
DWORD    dwReserved2;
DWORD    dwReserved3;
DWORD    dwSVBCaps;
DWORD    dwSVBCKeyCaps;
DWORD    dwSVBFXCaps;
DWORD    dwSVBRops[DD_ROP_SPACE];
DWORD    dwVSBCaps;
DWORD    dwVSBCKeyCaps;
DWORD    dwVSBFXCaps;
DWORD    dwVSBRops[DD_ROP_SPACE];
DWORD    dwSSBCaps;
DWORD    dwSSBCKeyCaps;

DWORD    dwSSBCFXCaps;
DWORD    dwSSBRops[DD_ROP_SPACE];
DWORD    dwReserved4;
DWORD    dwReserved5;
DWORD    dwReserved6;

} DDSCAPS, FAR* LPDDSCAPS;

```

DirectDraw  
DirectDrawSurface

DDSCAPS

IDirectDraw2::GetCaps

IDirectDrawPalette::GetCaps

dwSize

“;,, ,fff, “;Ž,,,‘ ’,,,,,,,,,”

---

**dwCaps**

ffff“”—,,,,fff

**DDCAPS\_3D**

ffffff f ffff,,,ffffff fff,Ž,,,,,,\*,

**DDCAPS\_ALIGNBOUNDARYDEST**

DirectDraw,ffff“E<E, •, dwAlignBoundaryDest  
,” ,,, ,,,ff f,,,,,\*

**DDCAPS\_ALIGNBOUNDARYSRC**

DirectDraw,ffff“E<E, •, dwAlignBoundarySrc  
,” ,,, ,,,ff f,,,,,\*

**DDCAPS\_ALIGNSIZEDEST**

DirectDraw  
,fff^,,Ž•E,‘,,,dwAlignSizeDest,” ,,, ,ffff“E<E,,,ff f,,,,,\*

**DDCAPS\_ALIGNSIZESRC**

DirectDraw  
,fff^,,•E,‘,,,dwAlignSizeSrc,” ,,, ,ffff“E<E,,,ff f,,,,,\*

**DDCAPS\_ALIGNSTRIDE**

DirectDraw ,dwAlignStrideAlign  
,“,,,ffffffŠ,Ž,,,,ffffff fff f fff, ,,,,,\*,

**DDCAPS\_ALPHA**

ffff““ ,Š fffffff f ffff,ffff ffff,ff f,,,,,\*

**DDCAPS\_BANKSWITCHED**

ffffff f ffff,fff ,‘,,, fffffff fff,ffff ffff,” ,’%” ,,,,,,  
\*,

**DDCAPS\_BLT**

ffffff f ffff,ffff““ ,,,,,,\*

**DDCAPS\_BLTCOLORFILL**

ffffff f ffff,ffff““,ff fff,,,,,,\*,

**DDCAPS\_BLTDEPTHFILL**

ffffff f ffff,ffff““,Zffff “• ,,,,,,\*

**DDCAPS\_BLTFOURCC**

ffffff f ffff,ffff““,ff ff f•Š,,,,,,\*,

**DDCAPS\_BLTQUEUE**

ffffff f ffff,”Šffff“ ,,,,,,\*

**DDCAPS\_BLTSTRETCH**

ffff““ , fffffff f ffff, ’,,,,,\*

**DDCAPS\_CANBLTSYSTEMEM**

ffffff f ffff,ffff fff,,,,,ffff fff,,ffff“ ,,,,,,\*

**DDCAPS\_CANCLIP**

ffffff f ffff,ffff““,ffffff,,,,,\*

**DDCAPS\_CANCLIPSTRETCHED**

```

        ffffff f ffff, 'ffff"',fffff,,,,,,,,*,
DDCAPS_COLORKEY
        f f ff,,ffff"',,,,,,,,,ff f ,,,,,,ĚŽ,ff f,, ",ff f"— •,
        dwCKeyCaps fff,,
DDCAPS_COLORKEYHWASSIST
        ff f ,f ffff,,ff f,,,,,,,,*,
DDCAPS_GDI
        ffffff f ffff, GDI ,<—,,,,,,,,*,
DDCAPS_NOHARDWARE
        f ffff ff f,,,,,,,,*,
DDCAPS_OVERLAY
        ffffff f ffff,f f ff,ff f,,,,,,,,*,
DDCAPS_OVERLAYCANTCLIP
        ffffff f ffff,f f ff,ff f,, ffffff,,,,,,,,*,
DDCAPS_OVERLAYFOURCC
        f f ff f ffff,f f ff —,Š ff ff f*Š,,,,,,,,*,
DDCAPS_OVERLAYSTRETCH
        f f ff f ffff, ',"—,,,,,,,,*,
DDCAPS_PALETTE
        DirectDraw ,fffff f fff,,,,, ',f fff,, DirectDrawPalette
        ffffff, ,, ff f,,,,,,,,Ž,
DDCAPS_PALETTEVSYNC
        DirectDraw,VSYNC,"Š,ffff, ,,,,,*,
DDCAPS_READSCANLINE
        ffffff f ffff,ffffff"Ž,,%"",,,,,*,
DDCAPS_STEREOVIEW
        ffffff f ffff,ffff ffff"—,Ž,,,,,,,,*,
DDCAPS_VBI
        ffffff f ffff,VffffŠ, ,, ,,,,,*,
DDCAPS_ZBLTS
        ffff"', ,,,Zffff,Ž—,ff f,,
DDCAPS_ZOVERLAYS
        ,, Ě,,f f ff,',',,IDirectDrawSurface2::UpdateOverlayZOrderf
        fff,Ž—,ff f,,

dwCaps2
        ,,ffff"',—, ,*,fff
DDCAPS2_CERTIFIED
        ffffff f ffff,• ,,,,,*,
DDCAPS2_NO2DDURING3DSCENE
        IDirect3DDevice::BeginSceneffffĚ, ,, IDirect3DDevice::EndScene
        ffffĚ, ,,Š, Direct3D,Ž—
    
```

---

,,,,,,f fff , IDirectDrawSurface2::Blt,,IDirectDrawSurface2::  
Lock,,, 2D —,Ž ,,,,,,\*,

**dwCKeyCaps**

ff f ”—,•,fff

**DDKEYCAPS\_DESTBLT**

ff f ,,“%Œffff“ ,ff f,, ff f ,RGB  
,ffff“ f fff,Š%”,fff,Ž,

**DDKEYCAPS\_DESTBLTCLRSPACE**

ff ff f,,“%Œffff“ ,ff f,, ff ff f,RGB  
,ffff“ f fff,Š%”,fff,Ž,

**DDKEYCAPS\_DESTBLTCLRSPACEYUV**

ff ff f,,“%Œffff“ ,ff f,, ff ff f,YUV  
,ffff“ f fff,Š%”,fff,Ž,

**DDKEYCAPS\_DESTBLTYUV**

ff f ,,“%Œffff“ ,ff f,, ff f ,YUV  
,ffff“ f fff,Š%”,fff,Ž,

**DDKEYCAPS\_DESTOVERLAY**

ff f fff,,f f ff,ff f,, ff f ,RGB  
,f f ff,,ffff“ f fff,Š%”,fff

**DDKEYCAPS\_DESTOVERLAYCLRSPACE**

RGB ,ffff“ ff f ,,ff ff f,ff f,,

**DDKEYCAPS\_DESTOVERLAYCLRSPACEYUV**

YUV ,ffff“ ff f ,,ff ff f,ff f,,

**DDKEYCAPS\_DESTOVERLAYONEACTIVE**

%Ž ‘,f f ff f fff, ,,,,—Œ,ffff“ ff f ’,ff f,,

**DDKEYCAPS\_DESTOVERLAYYUV**

ff f fff,,f f ff,ff f,, ff f ,YUV ,f f ff,,ffff“  
f fff,Š%”,fff

**DDKEYCAPS\_NOCOSTOVERLAY**

f f ff,ff f ,Ž—,,,,‘, —“,%Œ<,,,,\*,

**DDKEYCAPS\_SRCBLT**

RGB f fff,ffff“Œff f ,,“%Œffff“ ,ff f,,

**DDKEYCAPS\_SRCBLTCLRSPACE**

RGB f fff,ffff“Œff ff f,,“%Œffff“ ,ff f,,

**DDKEYCAPS\_SRCBLTCLRSPACEYUV**

YUV f fff,ffff“Œff ff f,,“%Œffff“ ,ff f,,

**DDKEYCAPS\_SRCBLTYUV**

YUV f fff,ff f ,,“%Œffff“ ,ff f,,

**DDKEYCAPS\_SRCOVERLAY**

RGB ,ffff“Œff f ,,f f ff,ff f,,

**DDKEYCAPS\_SRCOVERLAYCLRSPACE**

```

    RGB
    f f ff f fff,ffff“Œff f ,,ff ff f,Ž,,f f ff,ff f,
    ,
DDCKEYCAPS_SRCOVERLAYCLRSPACEYUV
    YUV f f ff f fff,ffff“Œff f ,,ff ff f,ff f,,
DDCKEYCAPS_SRCOVERLAYONEACTIVE
    %Ž ‘,f f ff f fff, ,,,,—Œ,ffff“Œff f ’,ff f,,
DDCKEYCAPS_SRCOVERLAYYUV
    YUV f f ff f fff,ffff“Œff f ,,f f ff,ff f,,

dwFXCaps
    ffff“, ’,,FX”—
DDFXCAPS_BLTARITHSTRETCHY
    ffff,, ,,,Š‘,,fffff,, , ffff““ ,Š,Š‘, f fff, ,,%Ž —,Ž—
    ,, ,,, ,Ž ‘, ,%o,, ,,,
DDFXCAPS_BLTARITHSTRETCHYN
    ffff,, ,,,Š‘,,fffff,, , ffff““ ,Š,Š‘,, f fff, ,,%Ž —,Ž—
    ,, ,,,Ž ‘, ,%o,, ,,, ” ×1,×2, ,,, ,Š‘, ,,, “ ,,,
DDFXCAPS_BLMIRRORLEFTRIGHT
    ffff““ ‘, %o““ ,,, ,ff f,,
DDFXCAPS_BLMIRRORUPDOWN
    ffff““ ‘, %o““ ,,, ,ff f,,
DDFXCAPS_BLTROTATION
    ffff““ ‘,“^,%o“,ff f,,
DDFXCAPS_BLTROTATION90
    ffff““ ‘,“,%o“,ff f,,
DDFXCAPS_BLTSHRINKX
    ,Ž • ,%o,,f fff,“^, ,,,,ff f,, ,fff,ffff““ ,,,—Œ,,,
DDFXCAPS_BLTSHRINKXN
    ,Ž • ,%o,,f fff, •,^ ×1/2 ×1/3, ,,, , ,,,,ff f,,
    ,fff,ffff““ ,,,—Œ,,,
DDFXCAPS_BLTSHRINKY
    ,Ž ‘ ,%o,,f fff,“^, ,,,,ff f,, ,fff,ffff““ ,,,—Œ,,,
DDFXCAPS_BLTSHRINKYN
    ,Ž ‘ ,%o,,f fff, •,^ ×1/2 ×1/3, ,,, , ,,,,ff f,,
    ,fff,ffff““ ,,,—Œ,,,
DDFXCAPS_BLTSTRETCHX
    ,Ž • ,%o,,f fff,“Š‘ ,,,,ff f,, ,fff,ffff““ ,,,—Œ,,,
DDFXCAPS_BLTSTRETCHXN
    ,Ž • ,%o,,f fff, ” ×1,×2, ,,, ,Š‘, ,,,,ff f,,
    ,fff,ffff““ ,,,—Œ,,,
DDFXCAPS_BLTSTRETCHY

```

---

,Ž ' ,%o,,f fff,"^,Š',,,,,ff f,, ,,fff,ffff"" ,,—Œ,,  
**DDFXCAPS\_BLTSTRETCHYN**

,Ž ' ,%o,,f fff, " ×1,×2, ,,, ,Š',,,,,ff f,,  
,,fff,ffff"" ,,—Œ,,

**DDFXCAPS\_OVERLAYARITHSTRETCHY**

ffff,, ,,Š',fffff,,,, ffff"" ,Š,Š', f fff, ,%oŽ —,Ž—  
,, ,Ž ' ,%o,, ,,,

**DDFXCAPS\_OVERLAYARITHSTRETCHYN**

ffff,, ,,Š',fffff,,,, ffff"" ,Š,Š', f fff, ,%oŽ —,Ž—  
,, ,Ž ' ,%o,, ,,, " ×1,×2, ,,, ,Š', ,,,

**DDFXCAPS\_OVERLAYMIRRORLEFTRIGHT**

%o""f f ff,ff f,,

**DDFXCAPS\_OVERLAYMIRRORUPDOWN**

%o""f f ff,ff f,,

**DDFXCAPS\_OVERLAYSHRINKX**

,Ž • ,%o,,f fff,"^, ,,,,,ff f,, ,,fff, DDSCAPS\_OVERLAY  
f fff,,—Œ,, ,,fff,f fff,"—,,Ž, ,%o",,,,,,\*,,,

**DDFXCAPS\_OVERLAYSHRINKXN**

,Ž • ,%o,,f fff, " ×1,×2, ,,, , ,,,,,ff f,,  
,,fff, DDSCAPS\_OVERLAY f fff,,—Œ,, ,,fff,f fff,"—  
,,Ž, ,%o",,,,,,\*,,,

**DDFXCAPS\_OVERLAYSHRINKY**

,Ž ' ,%o,,f fff,"^, ,,,,,ff f,, ,,fff, DDSCAPS\_OVERLAY  
f fff,,—Œ,, ,,fff,f fff,"—,,Ž, ,%o",,,,,,\*,,,

**DDFXCAPS\_OVERLAYSHRINKYN**

,Ž ' ,%o,,f fff, •, ^ ×1/2 ×1/3, ,,, , ,,,,,ff f,,  
,,fff, DDSCAPS\_OVERLAY f fff,,—Œ,, ,,fff,f fff,"—  
,,Ž, ,%o",,,,,,\*,,,

**DDFXCAPS\_OVERLAYSTRETCHX**

,Ž • ,%o,,f fff,"^,Š',,,,,ff f,, ,,fff, DDSCAPS\_OVERLAY  
f fff,,—Œ,, ,,fff,f fff,"—,,Ž, ,%o",,,,,,\*,,,

**DDFXCAPS\_OVERLAYSTRETCHXN**

,Ž • ,%o,,f fff, " ×1,×2, ,,, ,Š',,,,,ff f,,  
,,fff, DDSCAPS\_OVERLAY f fff,,—Œ,, ,,fff,f fff,"—  
,,Ž, ,%o",,,,,,\*,,,

**DDFXCAPS\_OVERLAYSTRETCHY**

,Ž ' ,%o,,f fff,"^,Š',,,,,ff f,, ,,fff, DDSCAPS\_OVERLAY  
f fff,,—Œ,, ,,fff,f fff,"—,,Ž, ,%o",,,,,,\*,,,

**DDFXCAPS\_OVERLAYSTRETCHYN**

,Ž ' ,%o,,f fff, " ×1,×2, ,,, ,Š',ff f,,  
,,fff, DDSCAPS\_OVERLAY f fff,,—Œ,, ,,fff,f fff,"—  
,,Ž, ,%o",,,,,,\*,,,

**dwFXAlphaCaps**

ffff"" ,fff"—

**DDFXALPHACAPS\_BLTALPHAEDGEBLEND**

ffff“Ĉff f .f fff,fffŽ,,ffff ffffffff,ff f,, ffff““ ,,,  
Ž—,,

**DDFXALPHACAPS\_BLTALPHAPIXELS**

ffff ff fff,ffff •,ff f,,  
ffff ff fff,,ffff •,fff “,, , , ,,,,,,, ffff’,%0,,,,,•“—,,  
ffff •, “,,,,, ,,,,Š‘“—’,,, ffff““ ,,,,Ž—,,

**DDFXALPHACAPS\_BLTALPHAPIXELSNEG**

ffff ff fff,ffff •,ff f,,  
ffff ff fff,,ffff •,fff “,, , , ,,,,,,, ffff’,%0,,,,,“—,,  
ffff •, “,,,,, ,,,,Š‘“—’,,, ,fff,DDCAPS\_ALPHA  
,fff,, ,Ĉ,Ž,,,,, ffff““ ,,,,Ž—,,

**DDFXALPHACAPS\_BLTALPHASURFACES**

ffff —f fff,ff f,, ffff —f fff,fff “,, , , ,,,,,,,“  
ffff’,%0,,,,,•“—,, ffff •, “,,,,, ,,,,Š‘“—’,,, ffff““ ,,,,Ž  
—,,

**DDFXALPHACAPS\_BLTALPHASURFACESNEG**

ffff ffff, ffff’,%0,,,,,“—,,,,,•,  
ffff ffff f f, “,, , , ,,,,,,, ffff •, “,,,,, ,,,,Š‘“—  
,’,,, ,fff,DDCAPS\_ALPHASURFACES ,fff,, ,Ĉ,fff,,,,,,  
ffff““ ,,,,Ž—,,

**DDFXALPHACAPS\_OVERLAYALPHAEDGEBLEND**

ffff“Ĉff f .f fff,fffŽ,,ffff ffffffff,ff f,, f f ff,,Ž  
—,,

**DDFXALPHACAPS\_OVERLAYALPHAPIXELS**

ffff ff fff,ffff •,ff f,,  
ffff ff fff,,ffff •,fff “,, , , ,,,,,,, ffff’,%0,,,,,•“—,,  
ffff •, “,,,,, ,,,,Š‘“—’,,, f f ff,,Ž—,,

**DDFXALPHACAPS\_OVERLAYALPHAPIXELSNEG**

ffff ff fff,ffff •,ff f,,  
ffff ff fff,,ffff •,fff “,, , , ,,,,,,, ffff’,%0,,,,,“—,,  
ffff •, “,,,,, ,,,,Š‘“—’,,, ,fff,DDCAPS\_ALPHAPIXELS  
,fff,, ,Ĉ,Ž,,,,, f f ff,,Ž—,,

**DDFXALPHACAPS\_OVERLAYALPHASURFACES**

ffff —f fff,ff f,, ffff —f fff,fff “,, , , ,,,,,,,  
ffff’,%0,,,,,•“—,, ffff •, “,,,,, ,,,,Š‘“—’,,, f f ff,,Ž  
—,,

**DDFXALPHACAPS\_OVERLAYALPHASURFACESNEG**

ffff ffff, ffff’,%0,,,,,“—,,,,,•,  
ffff ffff f f, “,, , , ,,,,,,, ffff •, “,,,,, ,,,,Š‘“—  
,’,,, ,fff,DDCAPS\_ALPHASURFACES ,fff,, ,Ĉ,fff,,,,,,  
f f ff,,Ž—,,

**dwPalCaps**

ffff”—,•,fff

**DDPCAPS\_1BIT**

---

ffffff,,fff,,,,,\*, ff f ff,2,, -,,  
**DDPCAPS\_2BIT**  
 ffffff,,fff,,,,,\*, ff f ff,4,, -,,  
**DDPCAPS\_4BIT**  
 ffffff,,fff,,,,,\*, ff f ff,16,, -,,  
**DDPCAPS\_8BIT**  
 ffffff,,fff,,,,,\*, ff f ff,256,, -,,  
**DDPCAPS\_8BITENTRIES**  
 ffffff,,fff,ff ffffff,Ž',,, ,ff ff, ,,DDPCAPS\_1BIT  
 DDPCAPS\_2BIT ,,,, DDPCAPS\_4BIT"—,Ž,,,Ž,, ,,-  
 \*f fff,ffff bpp -,fff,, ,—Œ,,, ,,,, -  
 ',,fff,, ffff““ f fff ,,bpp ffff,ffffff,,  
**DDPCAPS\_ALLOW256**  
 ,,ffff,256, -,,,,'<,,,,,Ž,  
**DDPCAPS\_PRIMARYSURFACE**  
 ffff,ffffff f fff,ffff,,,,,,\*, ffff,\* , DDPCAPS\_VSYNC  
 ”—,Ž' ff f,,,,, fffffff,' ,"%o,,  
**DDPCAPS\_PRIMARYSURFACELEFT**  
 ffff, 'ffffff f fff,ffff,,,,,,\*, ffff,\* ,  
 DDPCAPS\_VSYNC"—,Ž' ff f,,,,, fffffff,' ,"%o,,  
**DDPCAPS\_VSYNC**  
 ffff,fff ,ffffff,“Š,, ,,,,,\*,  
**dwSVCaps**  
 ffff ffff”—,\* ,fff  
**DDSVCAPS\_ENIGMA**  
 ffff f ffff,,ffff ff ,ŽŒ,,,,,,\*,  
**DDSVCAPS\_FLICKER**  
 Ž”ffffff,,ffff ff ,ŽŒ,,,,,,\*,  
**DDSVCAPS\_REDBLUE**  
 , ,ffff, %0,Š,”,3DŠ<,ffff ff ,ŽŒ,,,,,,\*,  
 ,,,%0'(ff ff f),,Ž‘,“ ,,,,,,  
**DDSVCAPS\_SPLIT**  
 \*Šfff f,,ffff ff ,ŽŒ,,,,,,\*,  
**dwAlphaBltConstBitDepths**  
 DDBD\_2 DDBD\_4,,, DDBD\_8 ffff-,2fff 4fff ,,,8fff,Ž,  
**dwAlphaBltPixelBitDepths**  
 DDBD\_1 DDBD\_2 DDBD\_4,,, DDBD\_8 ffff-,1fff 2fff 4fff  
 ,,,8fff,Ž,  
**dwAlphaBltSurfaceBitDepths**  
 DDBD\_1 DDBD\_2 DDBD\_4,,, DDBD\_8 ffff-,1fff 2fff 4fff  
 ,,,8fff,Ž,

**dwAlphaOverlayConstBitDepths**

DDBD\_2 DDBD\_4,,, DDBD\_8 ffff-,2fff 4fff ,,,8fff,Ž,

**dwAlphaOverlayPixelBitDepths**

DDBD\_1 DDBD\_2 DDBD\_4,,, DDBD\_8 ffff-,1fff 2fff 4fff  
,,,8fff,Ž,

**dwAlphaOverlaySurfaceBitDepths**

DDBD\_1 DDBD\_2 DDBD\_4,,, DDBD\_8 ffff-,1fff 2fff 4fff  
,,,8fff,Ž,

**dwZBufferBitDepths**

DDBD\_8 DDBD\_16 DDBD\_24,,, DDBD\_32 ffff-  
,1fff 16fff 24fff,,,32fff,Ž,

**dwVidMemTotal**

ffffff fff,“,—

**dwVidMemFree**

<,ffffff fff,—

**dwMaxVisibleOverlays**

%Ž ‘,f f ff, ‘,

**dwCurrVisibleOverlays**

%Ž ‘,f f ff,Œ ,

**dwNumFourCCCodes**

FourCC f f,

**dwAlignBoundarySrc**

ffff“ŒŒ, •,^

**dwAlignSizeSrc**

ffff“ŒŒ, ‘,,

**dwAlignBoundaryDest**

ffff“ <Œ, •,^

**dwAlignSizeDest**

ffff“ <Œ, ‘,,^

**dwAlignStrideAlign**

fffff —

**dwRops[DD\_ROP\_SPACE]**

fff —,ff f

**ddsCaps**

^“,”—,Ž, DDSCAPS “

**dwMinOverlayStretch and dwMaxOverlayStretch**

1000”,,, ‘Œ,f f ffŠ‘Œ ,,,, 1.3=1300

**dwMinLiveVideoStretch and dwMaxLiveVideoStretch**

1000”,,, ‘Œ,fff fffŠ‘Œ ,,,, 1.3=1300

**dwMinHwCodecStretch and dwMaxHwCodecStretch**

1000”,,, ‘Œ,f ffff f fffŠ‘Œ ,,,, 1.3=1300

**dwReserved1, dwReserved2, dwReserved3**

—,,”,—,,”,

---

```

dwSVBCaps
    ffff fff fffffff ffffffff“,ffff”—
dwSVBCKeyCaps
    ffff fff fffffff ffffffff“,ffffff f ĆŠ
dwSVBFXCaps
    ffff fff fffffff ffffffff“,ffffFXĆŠ
dwSVBRops[DD_ROP_SPACE]
    ffff fff fffffff ffffffff“,fff —ff f
dwVSBCaps
    fffff fff ffff ffffffff“,ffff”—
dwVSBCKeyCaps
    fffff fff ffff ffffffff“,ffffff f ĆŠ
dwVSBFXCaps
    fffff fff ffff ffffffff“,ffffFXĆŠ
dwVSBRops[DD_ROP_SPACE]
    fffff fff ffff ffffffff“,fff —ff f
dwSSBCaps
    ffff fff ffff ffffffff“,ffff”—
dwSSBCKeyCaps
    ffff fff ffff ffffffff“,ffffff f ĆŠ
dwSSBCFXCaps
    ffff fff ffff ffffffff“,ffffFXĆŠ
dwSSBRops[DD_ROP_SPACE]
    ffff fff ffff ffffffff“,fff —ff f
dwReserved4, dwReserved5, dwReserved6
    —,,,—,,,—,,

```

## DDCOLORKEY

```

typedef struct _DDCOLORKEY{
    DWORD dwColorSpaceLowValue;
    DWORD dwColorSpaceHighValue;
} DDCOLORKEY, FAR* LPDDCOLORKEY;

```

①

**IDirectDrawSurface2::GetColorKey**  
**IDirectDrawSurface2::SetColorKey**

①

```

dwColorSpaceLowValue
    ff f ,,,—,,,”^,%oĆ’
dwColorSpaceHighValue
    ff f ,,,—,,,”^, Ć’

```

## DDOVERLAYFX

```

typedef struct _DDOVERLAYFX{

```

```

        DWORD dwSize;
        DWORD dwAlphaEdgeBlendBitDepth;
        DWORD dwAlphaEdgeBlend;
        DWORD dwReserved;
        DWORD dwAlphaDestConstBitDepth;
union
{
        DWORD dwAlphaDestConst;
        LPDIRECTDRAW_SURFACE lpDDSAAlphaDest;
};
        DWORD dwAlphaSrcConstBitDepth;
union
{
        DWORD dwAlphaSrcConst;
        LPDIRECTDRAW_SURFACE lpDDSAAlphaSrc;
};
        DDCOLORKEY dckDestColorkey;
        DDCOLORKEY dckSrcColorkey;

        DWORD dwDDFX;
        DWORD dwFlags;
} DDOVERLAYFX, FAR *LPDDOVERLAYFX;

```

### IDirectDrawSurface2::UpdateOverlay

#### dwSize

“‘,, ,fff, “,Ž,,‘,’ ,,,,,,,,,,”

#### dwAlphaEdgeBlendBitDepth

ffff fff ffffffffĈ ,Ž’,,,,,Ž—,,fff “

#### dwAlphaEdgeBlend

ffff fff ffffffff,ffff,,Ž,,,’

#### dwReserved

—,Ž—,,,—,,,”

#### dwAlphaDestConstBitDepth

ffff“ ,ffff’ ,Ž’,,,,,Ž—,,fff “

#### dwAlphaDestConst

ffff“ ,ffff ffff,,Ž,,,’

#### lpDDSAAlphaDest

ffff“ ,ffff ffff,,Ž,,f fff,ffff

#### dwAlphaSrcConstBitDepth

ffff“Ĉ,ffff’ ,Ž’,,,,,Ž—,,fff “

#### dwAlphaSrcConst

ffff“Ĉ,ffff ffff,,Ž,,,’

#### lpDDSAAlphaSrc

ffff“Ĉ,ffff ffff,,Ž,,f fff,ffff

#### dckDestColorkey

— ,ffff“ ff f

#### dckSrcColorkey

— ,ffff“Ĉff f

---

### dwDDFX

*f f ff EXfff*

### DDOVERFX\_ARITHSTRETCHY

*„Š‘,,, ,Ž,%o,,%oŽŠ‘,Ž—,,*

### DDOVERFX\_MIRRORLEFTRIGHT

*%o”“, ,*

### DDOVERFX\_MIRRORUPDOWN

*%o”“, ,*

### dwFlags

*„fff,Ā Ž,,,,, „fff,,,,,,,,,*

## DDPIXELFORMAT

```
typedef struct _DDPIXELFORMAT{
    DWORD dwSize;
    DWORD dwFlags;
    DWORD dwFourCC;
union
{
    DWORD dwRGBBitCount;
    DWORD dwYUVBitCount;
    DWORD dwZBufferBitDepth;
    DWORD dwAlphaBitDepth;
};
union
{
    DWORD dwRBitMask;
    DWORD dwYBitMask;
};
union
{
    DWORD dwGBitMask;
    DWORD dwUBitMask;
};
union
{
    DWORD dwBBitMask;
    DWORD dwVBitMask;
};
union
{
    DWORD dwRGBAlphaBitMask;
    DWORD dwYUVAAlphaBitMask;
};
} DDPIXELFORMAT, FAR* LPDDPIXELFORMAT;
```

① **IDirectDrawSurface2::GetPixelFormat**  
①①

DirectDrawSurface

**dwSize**

“‘,, ,fff, “Ž,,‘,’ ,,,,,,,,,,,

**dwFlags**

ffff 0fff

**DDPF\_ALPHA**

ffff ff fff,ffff —f fff,< ,,,

**DDPF\_ALHAPIXELS**

f fff,ffff ffff •Ž,,ffff ff fff,,

**DDPF\_COMPRESSED**

f fff,Ž’,,,ff fff,ffff f f,Ž,“ ,,, —,f f^ ,,

**DDPF\_FOURCC**

FourCC f f,—0,,,

**DDPF\_PALETTEINDEXED1****DDPF\_PALETTEINDEXED2****DDPF\_PALETTEINDEXED4****DDPF\_PALETTEINDEXED8**

f fff,,fff ,fff ,fff,,,,fff,ffffff,.,.,.

**DDPF\_PALETTEINDEXEDTO8**

f fff,,fff ,fff,,,fff, ,,fff ffff,ffffff,.,.,.

**DDPF\_RGB**

ffff ff fff “,,RGB f f,—0,,,

**DDPF\_RGBTOYUV**

f fff,RGB f f,Ž,“ ,,, ,,, —,Š,YUV

f f,•Š,, ,,,f f,ff fff,ffff ff fff “,Š,, DDPF\_RGB  
fff,,fff,.,.,.,.,.

**DDPF\_YUV**

ffff ff fff “,,YUV f f,—0,,,

**DDPF\_ZBUFFER**

Zffff —f fff,ffff ff fff,< ,,,

**dwFourCC**

FourCC f f

**dwRGBBitCount**

ffff—, RGB fff 4,8,16,24,.,.,32

**dwYUVBitCount**

ffff—, YUV fff DDBD\_4 DDBD\_8 DDBD\_16 DDBD\_24  
.,., DDBD\_32

**dwZBufferBitDepth**

Zffff fff “ 8,16,24,.,.,32

**dwAlphaBitDepth**

ffff ffff fff “ DDBD\_1 DDBD\_2 DDBD\_4 ,,, DDBD\_8



```

FFERf fff,,' ,f fff,,,,fff,Ž,,,,
'f fff, DDSCAPS_FLIP DDSCAPS_BACKBUFFER
fff,' ,,, DDSCAPS_FRONTBUFFER
,Œ”,,fff ffff,,”,, ,”—
, IDirectDraw2::CreateSurfaceffff,',,,, “—Œ,fff ffff, ,,,
,,ffff,Œ, ,,,Œ ,f fff, f fff ffffff “,Œ ,,,,ffff ffff
,,,,,fff ffff ,,,—*,,ffff,,,,,,, ,
IDirectDrawSurface2::AddAttachedSurface,Ž ,,,, DirectDraw
,ffffff “,”, ,f fff,ff f,,

```

**DDSCAPS\_COMPLEX**

- f fff,< ,,,,,,,\*, • f fff,^ ,f fff, ,,,,,
- %of fff,f f f fff,ffff,, • “,,f f,”Š,,,,,,,”Š,,,,,,,

**DDSCAPS\_FLIP**

```

,,f fff,f fffffff “,^,,,,,,*, ,”—
, IDirectDraw2::CreateSurfaceffff,“,,,, ,,,,ffff ffff,,,,,^ ,fff
ffff, ,,, DirectDraw
,ffff ffff,DDSCAPS_FRONTBUFFERfff,,ffff ffff,—
,,,f fff, DDSCAPS_BACKBUFFERfff,fff,,
,fff,Ž’,, DDSURFACEDESC “,
dwBackBufferCountfff, ,,,,,,fff,,,,,,,
IDirectDraw2::CreateSurfaceffff,Ž—
,, ‘ ,f fff, ,Ž ,DDSCAPS_COMPLEXfff,fff,,,,,,,

```

**DDSCAPS\_FRONTBUFFER**

```

,,f fff,f fff fffff “,ffff ffff,,,,,,*,
,,fff,^”, DDSCAPS_FLIP ”—
,fff,,,, IDirectDraw2::CreateSurfaceffff,,,,fff,,
IDirectDraw2::CreateSurfaceffff,Ž—,, “—Œfff ffff, ,,,
,f fff, DDSCAPS_FLIPfff,Ž,, ffffff “,Œ ,,,, IDirectDraw
Surface2::AddAttachedSurface ,Ž—,,,,,,,’fff ffff,ffff,,

```

**DDSCAPS\_HWCODEC**

```

,,f fff,f ffff,,^ %o ,,,fff f,Ž,,,,,,*,

```

**DDSCAPS\_LIVEVIDEO**

```

,,f fff,fff fff,Ž,Ž,,%o”,,,,,,*,

```

**DDSCAPS\_MIPMAP**

```

,,f fff,,fff,ffffff,,,,,,*,
,,f fff,ffffff,Œ ,,,,’,DDSCAPS_MIPMAP
f fff,ffff,, ,,, ‘ ,f fff, IDirectDrawSurface2::AddAttache
dSurfaceffff,Ž—,,,,,—Ž“ ,,,, ,,,,
IDirectDraw2::CreateSurfaceffff,Ž—,,,,,^—“ ,,, ,,,,fff,fff,,
DDSCAPS_TEXTURE,fff,,,,,,,

```

**DDSCAPS\_MODEX**

```

,,f fff, 320×200,, , 320×240,f fX10 (f f X) f fff,,,,,,*,

```

**DDSCAPS\_OFFSCREENPLAIN**

```

,,f fff,f f ff fffff Zffff ffff ffff fff ffff ,,,,ffff
f fff,,,,,, fff fŠ,f fff,,,,,,*, ff f f fff,Ž*,,,,,,Ž,,

```

**DDSCAPS\_OVERLAY**

```

,,f fff,f f ff,,,,,,*,
Œ ,f fff,f f ff,,,,,, , %oŽ,,,, ,,,,,,Œ,,

```

---

```

        DDSCAPS_VISIBLE , f fff,Ā f f ff,,,,,,,,,“”,,,,Ž,,,,,
DDSCAPS_OWNDC
        ,,f fff,ŽŠ,,,,ffff ffffff ,, ,,Š~Ž,,,,,,*,
DDSCAPS_PALETTE
        ,,ffff ffff,,^“,DirectDrawPalette
        ffffff, ,,f fff,ffff,,,,*,
DDSCAPS_PRIMARYSURFACE
        ,,f fff,fffff f fff,,,,*, ,,f fff,%Ž,,,,,^-,
DDSCAPS_PRIMARYSURFACELEFT
        ,,f fff, —,fffff f fff,,,,*, ,,f fff,f f, -,Ā,,,,“—,*,
        ,,f fff, ,,Ž
        DDSCAPS_PRIMARYSURFACEfff,Ž,,,,f fff, ^-,,,%—
        ,fffff f fff,%Ž,,
DDSCAPS_SYSTEMMEMORY
        ,,f fff fff,ffff fff,Š,“,,,,*,
DDSCAPS_TEXTURE
        ,,f fff,,fffff,,—,,,,,,*, ,, *,,,,fffff,,Ž,,,,,
DDSCAPS_VIDEMEMORY
        ,,f fff,fffff fff,‘ ,,,,,*,
DDSCAPS_VISIBLE
        ,,f fff,‘,, ,,* ,‘ ,%—,“%,,,,,,*, ,,fffff f fff,fff,,
DDSCAPS_WRITEONLY
        f fff, ,, —,,,,,,*, f fff,,“, ,ffff,“”Ā ,, —
        Š, ,,%” ,, ,,f fff,,“, ,Ā%“,“—,“ ,,,
DDSCAPS_ZBUFFER
        ,,f fff,Zffff,,,,,,*, Zffff,Ž,,,, *Š,,,, ffff,%— <“,——
        ,,fff “ *Š,,,,

```

## DDSURFACEDESC

```

typedef struct _DDSURFACEDESC{
    DWORD dwSize;
    DWORD dwFlags;
    DWORD dwHeight;
    DWORD dwWidth;
    LONG lPitch;
    DWORD dwBackBufferCount;
    union
    {
        {
            DWORD dwMipMapCount;
            DWORD dwZBufferBitDepth;
            DWORD dwRefreshRate;
        }
    };

    DWORD dwAlphaBitDepth;
    DWORD dwReserved;

```

```

LPVOID      lpSurface;
DDCOLORKEY ddckCKDestOverlay;
DDCOLORKEY ddckCKDestBlt;

DDCOLORKEY ddckCKSrcOverlay;
DDCOLORKEY ddckCKSrcBlt;
DDPIXELFORMAT ddpfPixelFormat;
DDSCAPS     ddsCaps;
} DDSURFACEDESC, FAR* LPDDSURFACEDESC;

```

⑦

### IDirectDraw2::CreateSurface

#### dwSize

“,,, ,fff, “,Ž,,,’, ’ ,,,,,,,,,,”

#### dwFlags

ffff ☐fff ~%o,fff,,,,,,,,,• ,’ ,Ž, “,,,fff,—☐,,,Ž’,,

#### DDSD\_ALL

“—,,,,,,,,fff,—☐,,,,,,,,,•,

#### DDSD\_ALPHABITDEPTH

dwAlphaBitDepth fff,—☐,,,,,,,,,•,

#### DDSD\_BACKBUFFERCOUNT

dwBackBufferCount fff,—☐,,,,,,,,,•,

#### DDSD\_CAPS

ddsCaps fff,—☐,,,,,,,,,•,

#### DDSD\_CKDESTBLT

ddckCKDestBlt fff,—☐,,,,,,,,,•,

#### DDSD\_CKDESTOVERLAY

ddckCKDestOverlay fff,—☐,,,,,,,,,•,

#### DDSD\_CKSRCLT

ddckCKSrcBlt fff,—☐,,,,,,,,,•,

#### DDSD\_CKSRCOVERLAY

ddckCKSrcOverlay fff,—☐,,,,,,,,,•,

#### DDSD\_HEIGHT

dwHeight fff,—☐,,,,,,,,,•,

#### DDSD\_MIPMAPCOUNT

dwMipMapCount fff,—☐,,,,,,,,,•,

#### DDSD\_PITCH

lPitch fff,—☐,,,,,,,,,•,

#### DDSD\_PIXELFORMAT

ddpfPixelFormat fff,—☐,,,,,,,,,•,

#### DDSD\_REFRESHRATE

dwRefreshRate fff,—☐,,,,,,,,,•,

---

**DDSD\_WIDTH**

**dwWidth** fff,—E,,,,,•,

**DDSD\_ZBUFFERBITDEPTH**

**dwZBufferBitDepth** fff,—E,,,,,•,

**dwHeight**

f fff, ,

**dwWidth**

“—f fff,•

**lPitch**

Ž,fffŠŽ,,,← •’,, fff

**dwBackBufferCount**

fff ffff,

**dwMipMapCount**

fffffffff,

**dwZBufferBitDepth**

Zffff, “

**dwRefreshRate**

ffffff“ fffffff f f,← ,,ŽŽ—,,

**dwAlphaBitDepth**

ffff ffff, “

**dwReserved**

—,,,,,

**lpSurface**

Š~,,,,,f fff fff,ffff

**ddckCKDestOverlay**

f f ff ,Ž—,,ff f

**ddckCKDestBlit**

ffff“ ,Ž—,,ff f

**ddckCKSrcOverlay**

f f ffE,Ž—,,ff f

**ddckCKSrcBlit**

ffff“E,Ž—,,ff f

**ddpfPixelFormat**

f fff,ffff ff fff<

**ddsCaps**

DirectDraw f fff”—

•’

*IDirectDraw2 IDirectDrawSurface2 IDirectDrawPalette  
IDirectDrawClipperIDirectDraw2*

∞∞∞

**DD\_OK**

—<, Ć,Š—,,

**DDERR\_ALREADYINITIALIZED**

ffffff,,,, Š%o,,,,

**DDERR\_BLTFASTCANTCLIP**

DirectDrawClipper

ffffff,IDirectDrawSurface2::BltFastffff,Ć, ,,,”,”“Ćf fff,ffff,,,,

**DDERR\_CANNOTATTACHSURFACE**

—<,,f fff,,f fff,ffff,,,

**DDERR\_CANNOTDETACHSURFACE**

—<,,f fff,,f fff,ffff,,,

**DDERR\_CANTCREATEDC**

Windows,,^ ,,,,ffff fffffff DC , ,,,,,,

**DDERR\_CANTDUPLICATE**

ffffff f fff 3Df fff ^~“, ,,f fff,• ,,,

**DDERR\_CANTLOCKSURFACE**

DCI ff f,,ffffff f fff,fff,,,,,,,,, ,,f fff,,ffff,“”,,

**DDERR\_CANTPAGELOCK**

f fff,f f fff,Ž”,,

f f fff,ffffff fff f fff,,,”fff f,,ffffff f fff, ,,“ ”,

**DDERR\_CANTPAGEUNLOCK**

f fff,f f fff%o ,Ž”,,

f f fff%o ,ffffff fff f fff,,,”fff f,,ffffff f fff, ,,“ ”,

**DDERR\_CLIPPERISUSINGHWND**

ffffff ffff,,,”fff ,,,DirectDrawClipper

ffffff,ffffff fff,fff,,,,,,,,

**DDERR\_COLORKEYNOTSET**

“Ćff f ,Ž’,,,,,

**DDERR\_CURRENTLYNOTAVAIL**

Ć ff f,,,,,

**DDERR\_DCALREADYCREATED**

ffff fffffff(DC),,,,,,f fff,‘,•,,,,, ^f fff,,,,,,,,,Ž“,,,,,,

**DDERR\_DIRECTDRAWALREADYCREATED**

DirectDrawffffff,,,” ,,,,,

**DDERR\_EXCEPTION**

—<,, — ,Š,—Š,” ,,

**DDERR\_EXCLUSIVEMODEALREADYSET**

,””“f f,fff,,,,,,,,, <’fff,fff,,,,,,,,

**DDERR\_GENERIC**



ffff“f ffff,‘ ,,,  
**DDERR\_NOCLIPLIST**  
 ffffff fff,—,,,,  
**DDERR\_NOCLIPPERATTACHED**  
 ,,f fff ffffff,,DirectDrawClipper ffffff,ffff,,,,  
**DDERR\_NOCOLORCONVHW**  
 •Šf ffff,‘ ,,, ,,,,—,,,,,, —,Ž ,,,,  
**DDERR\_NOCOLORKEY**  
 f fff,€ ff f ,Ž,,,,  
**DDERR\_NOCOLORKEYHW**  
 “ ff f ,‘,f ffffff f,,,,, —,Ž ,,,,  
**DDERR\_NOCOOPERATIVELEVELSET**  
 IDirectDraw2::SetCooperativeLevelffff,€, ,,,,, Š ,€, ,,,  
**DDERR\_NODC**  
 ,, ,,,,,,f fff—, ,,,,,,  
**DDERR\_NODDROPSHW**  
 DirectDraw fff — ROP f ffff,—,,,,  
**DDERR\_NODIRECTDRAWHW**  
 f ffff — DirectDraw ffffff ,,,, ffff,f ffff,ff f,,  
**DDERR\_NODIRECTDRAWSUPPORT**  
 € ,ffffff ffff,DirectDraw,ff f,,,,  
**DDERR\_NOEMULATION**  
 ffffff ffff fff,—,,,,  
**DDERR\_NOEXCLUSIVEMODE**  
 ffff fff,‘,““f f,—,,,, ffff fff,““f f,Ž,,,,  
**DDERR\_NOFLIPHW**  
 %Žf fff,ffffff,,,,  
**DDERR\_NOGDI**  
 GDI,‘ ,,,  
**DDERR\_NOHWND**  
 ffffff”,ffffff ffff,•—  
 ,, ,,,,ffffff ffff,‘fff fffff ffff,,,,,,,fff,,,,,,  
**DDERR\_NOMIPMAPHW**  
 ffffff fffff fffff,f ffff,‘ ,,,,,,—,,,,,, —,Ž ,,,,  
**DDERR\_NOMIRRORHW**  
 ff fff f ffff,‘ ,,,,,,—,,,,,, —,Ž ,,,,  
**DDERR\_NOOVERLAYDEST**  
 “ Š—  
 ,,,,IDirectDrawSurface2::UpdateOverlayffff€, ,,,,,,f f ff,‘, IDi  
 rectDrawSurface2::GetOverlayPositionffff,€, ,,,  
**DDERR\_NOOVERLAYHW**

---

*f f ff f ffff, ' ,,,,,,,,,,——, ,,,,, —,Ž ,,,,*  
**DDERR\_NOPALETTEATTACHED**  
*ffff fffffff,,f fff,ffff, ,,,,,,*  
**DDERR\_NOPALETTEHW**  
*16 ,,,,256 ,ffff, ' ,f ffff ff f, ,,,*  
**DDERR\_NORASTEROPHW**  
*" ,fff —f ffff, ' ,,,,,,,,,,——, ,,,,, —,Ž ,,,,*  
**DDERR\_NOROTATIONHW**  
*%“f ffff, ' ,,,,, ,,,,——, ,,,,, —,Ž ,,,,*  
**DDERR\_NOSTRETCHHW**  
*Š‘f ffff, ' ,,,,,,,,,,——, ,,,,, —,Ž ,,,,*  
**DDERR\_NOT4BITCOLOR**  
*DirectDrawSurface fffffff, —<,,,fff ffff,Ž, ,,,,,*  
**DDERR\_NOT4BITCOLORINDEX**  
*DirectDrawSurface fffffff, —<,,,fff,ff fffffff ffff,Ž, ,,,,,*  
**DDERR\_NOT8BITCOLOR**  
*DirectDrawSurface fffffff, —<,,,fff ffff,Ž, ,,,,,*  
**DDERR\_NOTAOVERLAYSURFACE**  
*”f f ff f fff, ‘, f f ff fff fff,Ⓔ, ,,,*  
**DDERR\_NOTTEXTUREHW**  
*ffffff fffff f ffff, ‘ ,,,,,,,,,,——, ,,,,, —,Ž ,,,,*  
**DDERR\_NOTFLIPPABLE**  
*ffffff, ,,,f fff,ffffff, ,,,,,,*  
**DDERR\_NOTFOUND**  
*—<,,, —,Ⓔ, ,,,,,,*  
**DDERR\_NOTINITIALIZED**  
*ffffff, Š%,,, ‘, CoCreateInstance, ,,, DirectDraw  
ffffff,fff ffff ffff,Ⓔ, ,,,,,*  
**DDERR\_NOTLOCKED**  
*fff, ,,,,f fff,fff% , ,,,*  
**DDERR\_NOTPAGELOCKED**  
*- —,f f fff, ,,,f fff,f f fff% , ,,,*  
**DDERR\_NOTPALETTIZED**  
*Ž—, ,,,,f fff,ffff f f, f fff, ,,,,*  
**DDERR\_NOVSYNCHW**  
*’< “Š —, ‘, ,f fffffff f, ,,,, —,Ž ,,,,*  
**DDERR\_NOZBUFFERHW**  
*Zffff, ‘, ,f ffff ff f, ,,,, fffffff fff, Zffff ,,,, Zffff, Ž—  
,ffff“ ‘, Ž ,,,,*  
**DDERR\_NOZOVERLAYHW**

*f f ff f fff, f ffff,f f ff Zf f ,ff f,,,,, Zf f ,-Ž,,*  
**DDERR\_OUTOFCAPS**  
*—<,, —,\*—,f ffff,,,Š,“,,,,,*  
**DDERR\_OUTOFMEMORY**  
*DirectDraw, —, ,, \*fff,,*  
**DDERR\_OUTOFVIDEOMEMORY**  
*DirectDraw, —, ,, \*ffffff fff,,*  
**DDERR\_OVERLAYCANTCLIP**  
*f ffff,ffffff f f ff,ff f,,*  
**DDERR\_OVERLAYCOLORKEYONLYONEACTIVE**  
*f f ff ,,,^ ,—E,ff f ,Ž,,,,,*  
**DDERR\_OVERLAYNOTVISIBLE**  
**IDirectDrawSurface2::GetOverlayPosition**  
*ffff,\*%Žf f ff ,E, ,,*  
**DDERR\_PALETTEBUSY**  
*ffff,,,,,ffff,,,,,fff,,,,, ffff,<”,,*  
**DDERR\_PRIMARYSURFACEALREADYEXISTS**  
*,, —,,,,ffffff f fff, ,,,*  
**DDERR\_REGIONTOOSMALL**  
**IDirectDrawClipper::GetClipList***ffff,“,,,—^, ,,,*  
**DDERR\_SURFACEALREADYATTACHED**  
*,,ffff,,,,,,,f fff,f fff,ffff,,,,,*  
**DDERR\_SURFACEALREADYDEPENDENT**  
*,,‘f fff,^,,,,f fff,^,,,,,*  
**DDERR\_SURFACEBUSY**  
*f fff,,,,,ffff,,,,,fff,,,,, f fff,,ffff,<”,,*  
**DDERR\_SURFACEISOBSCURED**  
*f fff,,ffff, f fff,\*%o,,,,,,<”,,*  
**DDERR\_SURFACELOST**  
*f fff fff,\*,,, f fff,,ffff,<”,, ,f fff,\*DirectDrawSurface  
fffff,,IDirectDrawSurface2::Restoreffff,E, ,,,,,,*  
**DDERR\_SURFACENOTATTACHED**  
*—<,,f fff,ffff,,*  
**DDERR\_TOOBIGHEIGHT**  
*DirectDraw ,,,—<,, ,‘,,,*  
**DDERR\_TOOBIGSIZE**  
*DirectDraw ,,,—<,,‘,,‘,,, ,,, E\*, ,,\*—‘,,,*  
**DDERR\_TOOBIGWIDTH**  
*DirectDraw ,,,—<,,\*‘,,,*  
**DDERR\_UNSUPPORTED**

---

*ff f,,,,, —, ,*

**DDERR\_UNSUPPORTEDFORMAT**

*—, , FourCC ff fff, DirectDraw , , ff f, , , ,*

**DDERR\_UNSUPPORTEDMASK**

*—, , ffff ff fff, , ffffff , DirectDraw , , ff f, , , ,*

**DDERR\_UNSUPPORTEDMODE**

*ffffff, & ff f, , , , f f, ,*

**DDERR\_VERTICALBLANKINPROGRESS**

*' < ŠŠ, ,*

**DDERR\_WASSTILLDRAWING**

*, , f fff, , , f fff, , %0, ffff““ , Š—, , , ,*

**DDERR\_WRONGMODE**

*^, , f f, , , , , , f fff, ffff, , , , , , , ,*

**DDERR\_XALIGN**

*—, , , < & , • —, , , , ^ , • ^ —, • , , , , , , , , , ,*