

Microsoft[®] DirectX[™] 3

ffffffŠ"fff

„ffffff, < „„„ „• — „„• „„„„„ „ „• „%Ž- -
„„f f, < „„„„„„%<„„„„ „ffffff, „•, ^, Microsoft Corporation, -„„, <%
„„„ „< „Š„„„„„Ž •-„„• „„„„„„
Microsoft „„ffffff, < „„„„ ffff fff, <„„„ ‚, < • ‚ Ć „„, ‚“ —
Ć, •Ž„„„ „ffffff, Microsoft „-„„, <%„„„ „< • ‚ Ć „„, ‚“ —Ć, <%„„—
„„„„„

©1996 Microsoft Corporation. All rights reserved.

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

„„, •-„%Ž-, ŠŽ, •„„

DirectDraw

DirectDraw,,,,,.....
 DirectDrawf□fffff.....
 DirectDrawŠ—.....
 DirectDraw.....
 ,, ,DirectDraw,“.....
 DirectDraw HAL.....
 DirectDraw HEL.....
 DirectDrawffffff,fff.....
 •,fff.....
 3Df□fff,ff□f.....
 DirectDraw,Direct3D,“□.....
 f□fXffffff□f□f.....
 ffff□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.....
 □“%o,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—^,“Žffff

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

Windows 95 Windows NT® version 4.0
DirectDraw の

DirectDrawf□fffff

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 GDI

のの¥ の ¥のの Windows 95 DirectDraw Windows NT

GDI **CreateDIBSection**

DirectDraw

の DirectDraw DirectDrawPalette ¥

¥ の

IDirectDrawPalette

Windows ¥

DirectDrawPalette DirectDraw

DirectDraw DirectDrawClipper DirectDraw

¥の

„DirectDraw,“

<i>DirectDraw</i>	DirectDraw
-------------------	------------

[illegible]

DirectDraw HAL

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

DirectDraw HAL HAL
の
HAL HAL
DirectDraw

DirectDraw HEL

DirectDraw ¥ ¥ (HEL) HAL
DirectDraw DirectDraw HAL
DirectDraw

DirectDrawfffff,fff

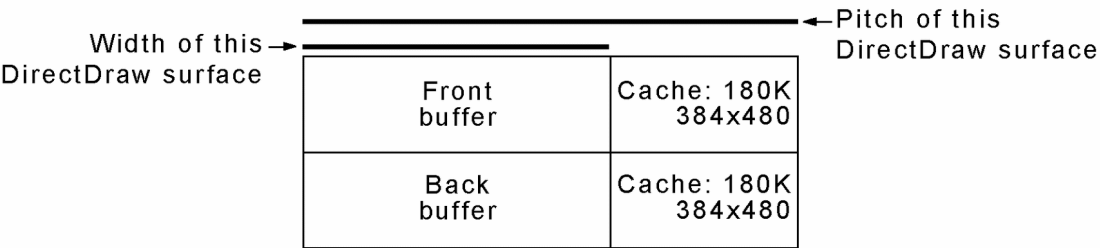
DirectDraw
DirectDraw DirectDraw
DirectDraw

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 ¥ 640×480×8 384×480×8 の
640 384 1024

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

3Df **fff,ff** **f**

の DirectDraw 3D

- fffff fff
- fffffff
- Zffff

ffffff¥fff

□□□□¥□□ HEL □
□□□□¥□□¥

DDSCAPS_TEXTURE **DDSURFACEDESC** の **IDirectDraw2::CreateSurface**
ddsCaps

HEL ¥ ¥のの

ffffff

DirectDraw ¥ ¥
のの

の

の

DirectDraw
のの

IDirectDraw2::CreateSurface **DDSURFACEDESC の**
DDSCAPS_MIPMAP
DDSCAPS_TEXTURE
IDirectDrawSurface2::AddAttachedSurface **IDirectDraw2::CreateSurface**

の 256×256 128×128 64×64 32×32 16×16 のの

```

DDSURFACEDESC ddsd;
LPDIRECTDRAW2 lpDD;
ZeroMemory(&ddsd, sizeof(ddsd));
ddsd.dwSize = sizeof(ddsd);
ddsd.dwFlags = DDSD_CAPS | DDSD_MIPMAPCOUNT;
ddsd.dwMipMapCount = 5;
ddsd.ddsCaps.dwCaps = DDSCAPS_TEXTURE |
    DDSCAPS_MIPMAP | DDSCAPS_COMPLEX;
ddsd.dwWidth = 256UL;
ddsd.dwHeight = 256UL;

ddres = lpDD->CreateSurface(&ddsd, &lpDDMipMap);
if (FAILED(ddres))
{
    .
    .
    .
}

```

IDirectDraw2::CreateSurface ののののの
のの

IDirectDraw2::CreateSurface の 1×1

DirectDrawSurface2::GetAttachedSurface **DDSCAPS の**
DDSCAPS_MIPMAP DDSCAPS_TEXTURE
のの

```

LPDIRECTDRAW2 lpDD;
DDSCAPS ddsCaps;

lpDDLevel = lpDDMipMap;
lpDDLevel->AddRef();
ddsCaps.dwCaps = DDSCAPS_TEXTURE | DDSCAPS_MIPMAP;
ddres = DD_OK;
while (ddres == DD_OK)
{
    // , , fff, □-
    .
    .
    .
    ddres = lpDDLevel->GetAttachedSurface(
        &ddsCaps, &lpDDNextLevel);
    lpDDLevel->Release();
    lpDDLevel = lpDDNextLevel;
}

```

DirectDraw の Z の DDBLT_DEPTHFILL の Z

Note %o ,',Ž ,%oŽ, 3Dffffff ffff,^',,

DirectDraw,Direct3D,“□

のDirectDraw Direct3D

- Direct3Dffff fff ffff
- Direct3D ffff fff ffff
- Direct3D fffff fff ffff
- DirectDraw HEL, Direct3D

Direct3Dffff□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 □□□□¥

Direct3D DirectDraw □□□□

Direct3D

DirectDraw
Direct3D

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 fXfffff 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

ffff ff ff
の ¥ (HEL)
¥ の

- fffff fff,ff fff
- fffff f f fff,ff fff

ffffff fff,ff fff
HEL のred green blue alpha
のの

ffff ff fff fff fff “ fff
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

①①	DirectDraw
----	------------

DDSCAPS TEXTURE

fffff f f f f f, f f f f

DirectX 3 HEL のred
のの

green

blue

alpha

<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	の	Direct3D

DirectDrawfff□ffffŠ—

- の DirectDraw COM

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

IDirectDraw2 f f f f f f f f

の IDirectDraw2

- DirectDraw f f f f f f f f
- IDirectDraw2, %o, , , , , , ?
- —, , , • DirectDraw f f f f f f f f
- %o “ 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 f f f f f f f f
- CoCreateInstance, —, , DirectDraw f f f f f f f f,

DirectDraw f f f f f f f f

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 f f f f f f f f, □ □
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**
 IDirectDraw2::Draw
 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
〇〇

IDirectDraw
IDirectDraw2

□ — , , , • □ **DirectDrawffff**
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

DirectDraw HEL 24- bpp 32-bpp の

¥

HEL

Windows

DirectDraw

の

IDirectDraw2::SetDisplayMode

IDirectDraw2::EnumDisplayModes

の

fffff□f□fff¥ff□f←fff

DirectDraw

の

DirectDraw GDI の

GDI

GDI

の

DirectDraw GDI

GDI の DirectDrawSurface □□□□□□

IDirectDraw2::GetGDISurface

GDI DirectDraw ¥
(DDB)

DirectDraw の

DirectDraw の

HAL DIB

¥

f□f•□,”“ffff

IDirectDraw2::SetDisplayMode

¥

の

DirectDraw □□□□□□

DirectDrawSurface □□□□□□ ¥

IDirectDrawSurface2::Restore

DirectDraw

DirectDrawSurface □□□□□□

DirectDraw

GDI の

CoCreateInstance,—,DirectDrawffffff,□□

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
fff,Ž,,DirectDrawffffff, ,
ddrval = CoCreateInstance(&CLSID_DirectDraw,
    NULL, CLSCTX_ALL, &IID_IDirectDraw2, &lpdd);
if(!FAILED(ddrval))
    ddrval = IDirectDraw2_Initialize(lpdd, NULL);

,,CoCreateInstance,€, ,',fff fCLSID_DirectDraw,
DirectDrawffff fffffff fff,fffŽ•Ž,,, IID_IDirectDraw2
fff f, ,,“,,DirectDraw fff ffff,Ž•,,
lpddfff f, Ž“,,,DirectDrawffffff,Ž, €, ,, €, ,, ,Š , Š%o
,,,,ffffff,•,

3 DirectDrawffffff,Ž,‘, IDirectDraw2::Initialize,€, ,,,,,, ,ffff,
DirectDrawCreate Š ,~“,Ž,ffff GUID fff f ,,, NULL ,^ ,Ž,
DirectDrawffffff Š%o€, DirectDrawCreate Š ,Ž,, ,,,“—,Ž—
%o•,,,,, DirectDrawffffff,Š~,ffff,Ž,‘,
IDirectDraw2::Initialize ffff,€, ,,,
DDERR_NOTINITIALIZEDff ,” ,. .
```

CoUninitialize

COM

CoUninitialize();

IDirectDrawClipperfff□ffff

IDirectDrawClipper

IDirectDrawClipper

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

fffffff□fff

DirectDraw DirectDrawClipper

のの

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,Z ffff,^
 “,,ffffff,,, ~‘,,E, ,, fffffff,Z ffff,%o<,—,,,,,,

ffff”^,DirectDrawClipperfffffff

DirectDraw

DirectDrawClipper

DirectDrawClipper DirectDraw

DirectDrawClipper

DirectDrawCreateClipper

DirectDraw

DirectDawClipper

DirectDraw

DirectDrawClipper

DirectDraw

IDirectDraw2::CreateClipper DirectDrawClipper
DirectDrawClipper DirectDraw

**CoCreateInstance,—
,,DirectDrawClipperfffff,□□**

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
- "ffffff f fff ,,ffff '
- ffff,<—
- ffff¥fff

DirectDrawPalettefffff

DirectDrawPalette 16 256
DirectDrawPalette

DirectDrawSurface

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

IDirectDrawPalette::GetEntries**IDirectDrawPalette::SetEntries**

の dwFlags

DirectDrawPalette

の

の

IDirectDrawPalette::SetEntries

DirectDrawPalette

DirectDrawSurface

DirectDrawPalette

"fffff f ffff,,ffff, f"

¥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

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

 \mathcal{O}

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

Windows Windows

8

- `ffffff f f,ffff ffff fff`
- `ffffff f f,,IDirectDraw::CreatePalette,CE, ,`
- `ffffff f f,IDirectDrawPalette::SetEntries,CE, ,`

ffff \square *f* \square *f*, *ffff* \square *ffff* \square *fff*

PALETTEENTRY ⑦ peFlags

IDirectDraw2::CreatePalette

IDirectDrawPalette::SetEntries peFlags

3

- Windows "ffff"
 ~", Windows, 0,,9 246,,255,,,ffff,—
 ffffffff ,,, ffff fff,ffff, ',• ,,,,Ź,,, ,,, ,Š,256 ffff
 ffff,GetSystemPaletteEntries Win32
 Š ,E, , ,,,,,, ffff fff,' ,,,,ffff,Ž ,•,E—",,
 ,,,, ffff f ff,—,,,ffff,•

```

—“,ffff fffffff,fff,,Ž’,, PC_EXPLICIT fff,PALETTEENTRY
“peFlagsfff ,Š”, ffff,fff,,•—
ffff,ffffff,,peRedfff,fff,, ,fff 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’,, ,fff 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	’

ffff□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;

```



```
{
    pPaletteEntry[index] = pPaletteEntry[index+1];
}
pPaletteEntry[25] = temp;

// ',□',,,□ffff□ffff□`'',ffff,,,'□
// •□,,ffff,,,'ffff',",,,,,,,,
ddrval = lpDDPal->SetEntries(
    0,                                // fff,ff,,,,,,,,
    10,                               // □□,ffff
    16,                               // ffff□
    & (pPaletteEntry[10])); // f□f,",,,,,
```

IDirectDrawSurface2

- *DirectDrawSurfaceffffff*
- *IDirectDrawSurface2,%0, ,,,,,,?*
- *f fff,*
- *ff f¥ffff,ffff*
- *ffffff f fff ,GDI,ff f¥f f*
- *f fff, Ž*
- *•Š,ff fff•Š*
- *ff ¥f*
- *Zf f f f ff*
- *ffffff f f fff,,,ffffff ffff*
- *ffff ¥ffff¥f fff,, ffff““*

DirectDrawSurface 2D の ¥

DirectDrawSurface DirectDraw

DirectDrawSurface

IDirectDraw2::CreateSurface DirectDrawSurface

¥ の ¥

DirectDraw DirectDrawSurface

の DirectDrawSurface

DirectDrawSurface ¥ の の CPU

DirectDrawSurface Windows ¥

DirectDrawSurface

¥ の GDI (HDC) DirectDrawSurface Windows GDI

¥ の GDI HDC

¥ HDC

IDirectDrawSurface2,%0,□,,,,,?

DirectX COM
IDirectDrawSurface2

IDirectDrawSurface
IDirectDraw::QueryInterface

```
LPDIRECTDRAWSURFACE lpSurf;  
LPDIRECTDRAWSURFACE2 lpSurf2;  
  
// ffff,□□.  
memset(&ddsd, 0, sizeof(ddsd));  
ddsd.dwSize = sizeof(ddsd);  
ddsd.dwFlags = DDSF_CAPS | DDSF_WIDTH | DDSF_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 fff,□□

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

```

DirectDraw の

DDSCAPS の **dwCaps** の**DDSCAPS_SYSTEMMEMORY**

¥

DDSCAPS_VIDEOMEMORY

DirectDraw

IDirectDraw2::CreateSurface

IDirectDraw2::CreateSurface

DDSCAPS_COMPLEX

の

DirectDraw の

IDirectDraw::Release

IDirectDrawSurface2::Restore

fff 3

の ¥

の

DDSURFACEDESC

□□□ (

ddsd)

¥

DDSURFACEDESC ddsd;

ddsd.dwSize = sizeof(ddsd);

// ,fff,-E,DirectDraw,',,,

ddsd.dwFlags = DDSD_CAPS | DDSD_BACKBUFFERCOUNT;

// ,,,,fff¥ffff,fff□f□fff,←,,

//

ddsd.ddsCaps.dwCaps = DDSCAPS_COMPLEX | DDSCAPS_FLIP |

DDSCAPS_PRIMARYSURFACE;

ddsd.dwBackBufferCount = 1;

IDirectDrawSurface2::Flip

¥

の

DDSURFACEDESC □□□の

BackBufferCount □□□ 2

2 の

IDirectDrawSurface2::Flip

3 の

ff□f¥ffff,ffff

DirectDraw

DirectDrawSurface

IDirectDrawSurface2::Lock RECT □□□

IDirectDrawSurface2::Lock

RECT □□□ NULL

IDirectDrawSurface2::Lock

DDSURFACEDESC □□□

の

DirectDrawSurface

の

• ^,ffffff fff,',,,,,, ,,,**IDirectDrawSurface2::Lock**

ffff,*,fff *,',,,, fff, f fff fff, fffffff¥f f,fff

DirectDrawffff,f fff,,,,,,——,*,%o,,

- **IDirectDrawSurface2::Lock**, **IDirectDrawSurface2::Unlock**
`ffff,Š,,Ž,, Ą,,, f fff fff,ffffŽ“,^“, ,,,`
IDirectDrawSurface2::Lock
`ffff,WIN16,fff,•Ž,,, IDirectDrawSurface2::GetDC`
`ffff,^–,IDirectDrawSurface2::Lock,Ą, , WIN16fff,`
IDirectDrawSurface2::Lock, **IDirectDrawSurface2::Unlock**, Š, GDI, USE
`R,ffff,IDirectDrawSurface2::GetDC, IDirectDrawSurface2::ReleaseDC`
`,Š“,,,, Ą,,`
- `ffffff¥fff,~“,,,ff ,,,`
`fff¥ffff¥fff,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`
`¥fff,ff ,, ff ,ffffff,,,,, , ffff,^Ž’Ž,^,,‰” ,,,`

ffffff□f□fff ,GDIff□f¥f□f

DirectDraw のの

¥

DirectDraw

のの

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 fff),'Šff¥ffff,ffff“,,
3 GDI,‘, ,',, Šff ,%ž%,,,, lpDDSTargetOverride
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 dwRBits dwGBits dwBBits dwRGBAlphaBits
RGB
DDPF_RGB DDPF_FOURCC

FOURCC のののの FOURCC

ff ¥f fff

DirectDraw の¥

のの

¥

¥

¥

¥

¥のの¥

¥

YUV

YUV

¥

¥

¥の¥

¥の YUV

FOURCC

YUV

¥

DDCOLORKEY □□□. dwColorSpaceLowValue

dwColorSpaceHighValue □□□□□□□ 3 □□□

V

U

Y

IDirectDrawSurface2::SetColorKey

dwFlags

¥ののの¥の

8

```
// ffffffff26,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 fffZf□f□

Z □□□□

DirectDraw

¥

¥

¥のの

¥のの

¥

Z

の Z Z Z 0 Z Z
Z 2 Z 0 1 の Z 40 Z

fffff f f ffff,,,. f ffff

DirectDraw ののの¥ 16 のの

DirectDraw 256 8 4
16 の 4
のののの
のの
のののの

のののの¥の4 8

3D

¥ ¥のの ¥

ffff“ ,ffff ffff f ffff

DMA

¥の DDCAPS

12

dwSVBCaps	dwVSBCaps	dwSSBCaps
dwSVBCKeyCaps	dwVSBCKeyCaps	dwSSBCKeyCaps
dwSVBFXCaps	dwVSBFXCaps	dwSSBFXCaps
dwSVBRops	dwVSBRops	dwSSBRops

SVB	¥	¥	□□□□□□
VSB	¥		□□□□□□
SSB			□□□□□□

dwSVBCaps □□□	dwCaps □□□		
□□□□□□	¥	の	
dwSVBCKeyCaps □□□	dwCKeyCaps □□□	dwSVBFXCaps	□□□□
□□□ dwFXCaps □□□	dwSVBRops □□□		
□□			

dwCaps DDCAPS_CANBLTSYSMEM
 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*ff*
ffff¥*fff*,,,,this*ffff*,%,•—
 ,,, ,,, , , •, C,Ž,,COM*ffffff*,*ffff* ,Ž ,,,,

Tutorial 1: DirectDraw,Š‘

DirectDraw
 DirectDraw

DirectDrawSurface
 SDK
 DDEX1 のの

- Step 1: DirectDraw*ffffff* ,

- Step 2: `ffff fff, ,*,,OE'`
- 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,S,“,,%o•`

Step 1: `DirectDrawffffff,□□`

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

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

Step 2: `ffff□fff,□,*,,OE'`

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□f,□E,,.
}
else
{
    // "``f□f,ž",,
    // ,,□ffff□fff,,,",,,,
}

```

IDirectDraw::SetCooperativeLevel DD_OK

IDirectDraw::SetCooperativeLevel

Windows	¥	(HWND)
(GP)	GDI	¥
Windows		

DirectDraw

DirectDraw



¥

GDI

ALT+TAB

Step 3: $fffff \square f \square f, \bullet \square$

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,,,,,,□`,fffff□fff,
    // "``f□f,,,,,,
}

```

のの640×480×8 のの

(
IDirectDraw::SetDisplayMode	DDERR_INVALIDMODE

IDirectDraw::EnumDisplayModes)

Step 4: *fffff* *f* *fff,*

DDEX1

IDirectDraw::SetCooperativeLevel DDSCL_NORMAL
IDirectDraw::SetCooperativeLevel

- *f fff*—<,'<
- *f fff*

の

の

DDSURFACEDESC の

の

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

ddsd.dwBackBufferCount = 1;
```

の

dwSize の **DDSURFACEDESC** の

DirectDraw

dwSize の

DDSURFACEDESC のののの

dwFlags の

DDSURFACEDESC のの DDEX1 の

dwFlags

DDSCAPS の

(DDSD_CAPS)

(DDSD_BACKBUFFERCOUNT)

dwCaps の

DDSCAPS の

(DDSCAPS_PRIMARYSURFACE)

(DDSCAPS_FLIP)

(DDSCAPS_COMPLEX)

の

¥

¥

¥

DDEX1 のの 1

¥

¥

.

¥

DirectDraw

¥

(1MB RAM

¥

) **DDSCAPS** の **dwCaps** の

DDSCAPS_SYSTEMMEMORY

DDSCAPS_VIDEOMEMORY

¥

DDSCAPS_VIDEOMEMORY

IDirectDraw::CreateSurface

DDERR_OUTOFVIDEOMEMORY

f fff,□□

DDSURFACEDESC のののの

DirectDrawCreate

DirectDraw

lpDD

DDSURFACEDESC のののの

IDirectDraw::CreateSurface

¥ の
IDirectDrawSurface::GetDC Windows ¥
IDirectDrawSurface::Lock and **IDirectDrawSurface::Unlock**

¥の¥ ¥

の Windows GDI **SetBkColor**
SetTextColor **TextOut**

IDirectDrawSurface::ReleaseDC

IDirectDrawSurface::ReleaseDC **IDirectDrawSurface::Unlock**

¥
DDEX1
DDEX1 のののの
DDEX1 WM_TIMER ¥

Note **IDirectDrawSurface::Unlock**,f fff,fff% ,œ f fff fff,,ffff,
-œ,, f fff fff,—
œ,ffff,“,,, **IDirectDrawSurface::Lock**, ,ž,,,,,,,,

Step 6: f fff,, ,,,

DDEX1 WM_TIMER

```
case WM_TIMER:
    // f fff,ffffff
    if(bActive)
    {
        if (lpDDSDBack->GetDC(&hdc) == DD_OK)
        {
            SetBkColor(hdc, RGB(0, 0, 255));
            SetTextColor(hdc, RGB(255, 255, 0));
            if(phase)
            {
                TextOut(hdc, 0, 0, szFrontMsg, lstrlen(szFrontMsg));
                phase = 0;
            }
            else
            {
                TextOut(hdc, 0, 0, szBackMsg, lstrlen(szBackMsg));
                phase = 1;
            }
            lpDDSDBack->ReleaseDC(hdc);
        }
    }
```

IDirectDrawSurface2::GetDC

¥ SetBkColor SetTextColor

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

IDirectDrawSurface::ReleaseDC**Step 7: *ffffff* *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 = lpDDSPPrimary->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;

```

DDLoadPalette DirectDraw \Sdk\Samples\Misc\Ddutil.cpp
DirectDraw の

の DDEX
samples の Ddutil.cpp
DDEX2 **DDLoadPalette** Back.bmp DirectDrawPalette
DDLoadPalette
DDEX2 *ape*

DDEX2 の DirectDrawPalette

```

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

```

IDirectDraw::CreatePalette *ddpal*
DirectDrawPalette **DDLoadPalette**

ape 2 4 16 256
IDirectDraw::CreatePalette *dwFlags*
dwFlags DDPCAPS_8BIT
の 256 4

Step 2: *ffff*,□'

IDirectDrawSurface::SetPalette

DirectDrawPalette *ddpal* □□□□□□□□□□
の

```
ddrval = lpDDSPPrimary->SetPalette(lpDDPal);
```

```
if(ddrval != DD_OK)
    // SetPaletteŽ"
```

IDirectDrawSurface::SetPalette
DirectDrawSurface

DirectDrawPalette

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□fff,fff,",
ddsd.dwSize = sizeof(ddsd);
ddsd.dwFlags = DDS_DHEIGHT | DDS_DWIDTH;
pdds->GetSurfaceDesc(&ddsd);
```

ddsd **DDSURFACEDESC** □□□

DirectDraw ののの

□□ DDS_DHEIGHT DDS_DWIDTH

IDirectDrawSurface::GetSurfaceDesc

DDEX2 480 640

DDCopyBitmap ¥

DDSURFACEDESC □
の

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


dwFlags `DDSCAPS_OFFSCREEN` **DDSCAPS** `DDSCAPS_480 640`
DDSURFACEDESC `IDirectDraw::CreateSurface`

`IDirectDraw::CreateSurface`

DDSCAPS `DDSCAPS_SYSTEMMEMORY`
DDSCAPS_VIDEOMEMORY

`DDSCAPS_VIDEOMEMORY`
`DDERR_OUTOFVIDEOMEMORY`

Step 2: `fffff` `f` `f` `fff,ffffff,f` `f`
DDEX3 Frntback.bmp
InitSurfaces
DDCopyBitmap Ddutil.cpp

```
// fffffff
```

Microsoft Paint Frntback.bmp のの
DDCopyBitmap 1 の 1 の(`lpDDStOne`) 2 の 2 の(`lpDDStTwo`)

Step 3: `fff` `ffff` `ffffff` `f` `f` `fff,ffff`
WM_TIMER
DDEX3

```
rcRect.left = 0;  
rcRect.top = 0;  
rcRect.right = 640;  
rcRect.bottom = 480;
```



```

if (phase)
{
    pdds = lpDDSTwo;
    phase = 0;
}
else
{
    pdds = lpDDSOOne;
    phase = 1;
}
while(1)
{
    ddrval = lpDDSDBack->BltFast(0, 0, pdds, &rcRect, FALSE);
    if (ddrval == DD_OK)
    {
        break;
    }
}

```

phase ¥ □□□□□□
 IDirectDrawSurface::BltFast
 ¥(0,0) □□□□□ rcRect
 RECT □□□□□□□□□□□□□□□□の
 ののの
 FALSE (0)
 IDirectDrawSurface::Blt IDirectDrawSurface::BltFast
 の¥ ¥
 ¥ のの IDirectDrawSurface::BltFast
 10 ¥
 ¥
 IDirectDrawSurface::BltFast IDirectDrawSurface::Blt
 ¥ ¥のの
 □□□□□□

Tutorial 4: *ff* ¥ *f* ¥, *ffffffffffff* ¥ *fff*

(DDEX3)の¥の
 (DDEX4) ののの
 の

DDEX4 All.bmp DDEX4 のの¥

¥

DDEX4

- Step 1: *ff* ¥ *f* ¥ , ' ¥

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

Step 1: $ff\text{ }¥f\text{ },\text{ }'$

DirectDraw の doInit
 の $\text{ }¥\text{ }$ $\text{ }¥\text{ }$ $\text{ }¥\text{ }$
 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(0, 0) ¥ の
DDSetColorKey **DDColorMatch** ¥
DDCOLORKEY のの dwColorSpaceLowValue のの
 dwColorSpaceHighValue のの
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, $\text{ }'\text{ }'$

DDEX4 の All.bmp
 updateFrame 3 3
 の Win32 **GetTickCount**
GetTickCount
 の
IDirectDrawSurface::BltFast
 (lpDDOne) ¥の¥ ¥
 ¥ DDEX4 ¥
 ののの **IDirectDrawSurface::Flip**

IDirectDrawSurface::BltFast*dwTrans*

DDBLTFAST_NOCOLORKEY

¥ *dwTrans*

DDBLTFAST_SRCCOLORKEY

¥ 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-,ffff,,
for(i=0; i<256; i++)
{
    torusColors[i] = 0;
}

// f□ffff,fff,□@”••,ffff,,□f□ff¥fff□
// @,,,,fffffff,<~,•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)
1 y ( row) IPitch
DDSURFACEDESC ののの
torusColors のの
true "*"ddsd.IPitch"
(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 □□□□□□

,,', DirectDrawffff

DirectDraw

DirectX SDK

のの

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

DirectDraw

Duel, Iklowns, Foxbear, Palette, Flip2d

□ “%o,ffffff

SDK

DirectDraw ののののの

- ffffff,ffff“‘, ‘Ž“
- ff fff,”,ffff““
- ffffff¥f ffff”—,Œ’
- ffffff¥fff,,ffffff,Š”
- ffff fffffff

ffffff,ffff“‘,□ ‘Ž“

IDirectDrawSurface2::Flip

□□□□□□□□□□

¥

のの

□□□□□

□

DDERR_WASSTILLDRAWING
DD_OK

SDK の IDirectDrawSurface2::Flip

IDirectDrawSurface2::Flip

DDERR_WASSTILLDRAWING

□□□□□□

¥

IDirectDrawSurface2::GetFlipStatus

の□□□□□□

DDERR_WASSTILLDRAWING

□□□□□□

```
while (lpDDSTexture->GetFlipStatus (DDGFS_ISFLIPDONE) ==
    DDERR_WASSTILLDRAWING);

// '\,ffffff,□-,,,, \,
// ffff□fff,,,,□\,fff,ž□,,,

ddrval = lpDDSPPrimary->Flip (NULL, 0);

□□□□□□
IDirectDrawSurface2::GetBltStatus
IDirectDrawSurface2::GetFlipStatus    IDirectDrawSurface2::GetBltStatus
```

```
ff□¥ffff,”,ffff““
                                ¥
                                IDirectDrawSurface2::Blt

                                DDBLT_COLORFILL
IDirectDrawSurface2::Blt
```

```
の¥の

DDBLTFX ddbltfx;

ddbltfx.dwSize = sizeof(ddbltfx);
ddbltfx.dwFillColor = 0;
ddrval = lpDDSPPrimary->Blt(
    NULL,          // "\□
    NULL, NULL,    // "\E<E
    DDBLT_COLORFILL, &ddbltfx);

switch(ddrval)
{
    case DDERR_WASSTILLDRAWING:
        .
        .
        .
    case DDERR_SURFACELOST:
        .
        .
        .
    case DD_OK:
        .
        .
        .
    default:
}
}
```

fffffff¥f ffff”—,œ’

DirectDraw の 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¥ffff,,fffff,š”

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

ffff¥ffffffffff

ののの¥の¥ ¥

```
// ,,,,ffff¥fffff,fffff□f□fff,□□
ddsd.dwSize = sizeof(ddsd);
ddsd.dwFlags = DDSD_CAPS | DDSD_BACKBUFFERCOUNT;
ddsd.ddsCaps.dwCaps = DDSCAPS_PRIMARYSURFACE |
    DDSCAPS_FLIP | DDSCAPS_COMPLEX;
ddsd.dwBackBufferCount = 2;
ddrval = lpDD->CreateSurface(&ddsd, &lpDDSPPrimary, NULL);
if(ddrval == DD_OK)
{
    // `^,ffff¥fffff,,ffff,ž”
    ddscaps.dwCaps = DDSCAPS_BACKBUFFER;
    ddrval = lpDDSPPrimary->GetAttachedSurface(&ddscaps,
        &lpDDSBBackOne);
    if(ddrval != DD_OK)
        // ,,,ff□fff□f,•ž
        // `”,ffff¥fffff,,ffff,ž”
    ddscaps.dwCaps = DDSCAPS_BACKBUFFER;
    ddrval = lpDDSPPrimary->GetAttachedSurface(&ddscaps,
```

DirectDrawClipper

- DD_OK DDERR_INVALIDPARAMS DDERR_OUTOFMEMORY

dwFlags

DDF_... 0, ...,

lpDDClipper

DirectDrawClipper, ...,

pUnkOuter

COM , NULL, ,

DirectDraw

DirectDrawClipper

DirectDraw DirectDrawClipper

DirectDraw

DirectDraw

DirectDrawClipper

IDirectDraw2::CreateClipper

IDirectDraw2::CreateClipper

DirectDrawEnumerate

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

DirectDraw

NULL GUID

GDI

- DD_OK DDERR_INVALIDPARAMS,

lpCallback

..., HAL..., ..., Callback, ...

lpContext

..., ..., ..., ..., ..., ..., ...

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..., ..., ..., ..., ...

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

lpDriverName
ffff–,Ž,•Ž—,ffff

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

EnumSurfacesCallback

HRESULT WINAPI lpEnumSurfacesCallback(
LPDIRECTDRAW2 lpDDSurface,
LPDDSURFACEDESC lpDDSurfaceDesc, LPVOID lpContext);

IDirectDrawSurface2::EnumAttachedSurfaces の

- ,’, , ,DDENUMRET_OK,•, Ž,, ,DDENUMRET_CANCEL,•,

lpDDSurface
,,f fff,ffff,,f fff,ffff

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

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

fnCallback

HRESULT WINAPI lpfnCallback(LPDIRECTDRAW2 lpDDSurface,
LPVOID lpContext);

IDirectDrawSurface2::EnumOverlayZOrders の

- ,’, , ,DDENUMRET_OK,•, Ž,, ,DDENUMRET_CANCEL,•,

lpDDSurface
f fff,f f ff,,,,,f fff,ffff

lpContext

Š ,Æ, ,,,,,,f ffffŠ ,“,,,,ffff fff,’<,, “,.,ffff

IDirectDraw2

DirectDraw

IDirectDraw2**IDirectDraw2***DirectDraw***IDirectDraw2**

fff,Š,“,

Compact**Initialize**

ffffff,

CreateClipper**CreatePalette****CreateSurface**

ffff,”—

GetCaps

ffffff¥f f

EnumDisplayModes**GetDisplayMode****GetMonitorFrequency****RestoreDisplayMode****SetDisplayMode**

ffffff¥ff ff

GetScanLine**GetVerticalBlankStatus**

,,‘

GetAvailableVidMem**GetFourCCCodes****WaitForVerticalBlank**

“ Š<, ’

SetCooperativeLevel

f fff

DuplicateSurface**EnumSurfaces****FlipToGDISurface****GetGDISurface**

COM

IUnknown

COM

IUnknown

COM

IUnknown

DirectDrawClipper DirectDrawSurface
IDirectDrawSurface2::Blt **IDirectDrawSurface2::BltBatch**
IDirectDrawSurface2::UpdateOverlay ⑦

DirectDraw DirectDrawClipper
DirectDrawCreateClipper

IDirectDrawSurface2::GetClipper **IDirectDrawSurface2::SetClipper**

IDirectDraw2::CreatePalette

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

DirectDrawPalette

- DD_OK Ž”,,,Ž,ff ,*,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_NOCOOPERATIVELEVELSET
DDERR_NOEXCLUSIVEMODE
DDERR_OUTOFCAPS
DDERR_OUTOFMEMORY
DDERR_UNSUPPORTED

dwFlags

Ž,fff,,, ,,,^ ,',Ž,

DDPCAPS_1BIT

ffffff,1fff ff f ff,,ffff,,,,,,

DDPCAPS_2BIT

ffffff,,fff ff f ff,,ffff,,,,,,

DDPCAPS_4BIT

ffffff,,fff ff f ff,,ffff,,Œ,,

DDPCAPS_8BITENTRIES

8ffffff fffffff,Ž ,ffffff,Ž, ,fff,
DDPCAPS_1BIT DDPCAPS_2BIT DDPCAPS_4BIT,Ž, ,f fff
f fff,8bpp, ,—
Œ,, Šff ffff, ,fff', 8bppffff,f fff,ffffff,Ž,

DDPCAPS_8BIT

ffffff,8fff ff f ff,,ffff,256Œ,,

DDPCAPS_ALLOW256

,,ffff,256Œ,ffff,,,',,,

lpColorTable

„DirectDrawPaletteffffff, Š%,,,,2 4 16
256 „,PALETTEENTRYŒ”—,ffff

lpDDPalette

„ffff, Œ,, •,, „DirectDrawPaletteffffff,,ffff,ffff

pUnkOuter

COM %œ“, —,ŒŠ ,,,, Œ , NULL, ’,,,ff ,,,

IDirectDraw2::CreateSurface

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

DirectDrawSurface

- Œ,, DD_OK Ž”,,Ž,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,,—œœ „,DDSURFACEDESC “„,ffff

lpDDSurface

Œ, „ Œ,, —Œ,DirectDrawSurfacefff,„, Š%œ
„,f fff,ffff,ffff

pUnkOuter

COM %œ“, —,ŒŠ ,,,, Œ , NULL, ’,,,ff ,,,

lpEnumModesCallback

Ⓔ, ' ,,,,Ⓔ,,“,Ⓔ,,,EnumModesCallbackŠ ,ffff

DDSURFACEDESC □□□の dwRefreshRate □□□

IDirectDraw::EnumDisplayModes

IDirectDraw2::SetDisplayMode

dwRefreshRate □□

□

IDirectDraw2::EnumDisplayModes

IDirectDraw2::GetDisplayMode

IDirectDraw2::SetDisplayMode

IDirectDraw2::RestoreDisplayMode

IDirectDraw2::EnumSurfaces

HRESULT EnumSurfaces(DWORD dwFlags, LPDDSURFACEDESC lpDDSD,

LPVOID lpContext, LPDDENUMSURFACESCALLBACK

lpEnumSurfacesCallback);

- Ⓔ,,, DD_OK Ž”,,,Ž,ff ,*,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

dwFlags

Ž,fff,,,,,,

DDENUMSURFACES_ALL

Š , ' ,f fff,,,—<,,

DDENUMSURFACES_CANBECREATED

Š , ' , , , , ,f fff,—<,,

DDENUMSURFACES_DOESEXIST

Š , ' , , , ,f fff,—<,,

DDENUMSURFACES_MATCH

f fff fffffff,Š“,,f fff,’,

DDENUMSURFACES_NOMATCH

f fff fffffff,Š“,,,,,f fff,’,

lpDDSD

—,,,f fff,’<,,DDSURFACEDESC “,,ffff

lpContext

—<,,,fff,“,,,ffff fff,’<,, “,,ffff

lpEnumSurfacesCallback

^, ' ,,,,Ⓔ,,“,Ⓔ,,,EnumSurfacesCallbackŠ ,ffff

DDENUMSURFACES_CANBECREATED

IDirectDraw

dwFlags

の

IDirectDraw2::FlipToGDISurface

HRESULT FlipToGDISurface();

GDI

- E,,DD_OK,Ž”,Ž,ff f f,Ž,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_NOTFOUND

GDI

¥

¥

¥

IDirectDraw2::GetGDISurface**IDirectDraw2::GetAvailableVidMem**HRESULT GetAvailableVidMem(LPDDSCAPS lpDDSCaps,
LPDWORD lpdwTotal, LPDWORD lpdwFree);

¥

ののの

- E,, DD_OK Ž”,Ž,ff ,Ž,
DDERR_INVALIDCAPS
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_NODIRECTDRAWHW

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

—,,,ffffff¥fff,‘—,Ž“,• ,ffff

lpdwFree

E ,ffffff,‘,—,Ž“,• ,ffff

*lpdwTotal**lpdwFree* NULL

C++ の

¥

¥

¥の

IDirectDraw2::GetAvailableVidMem

```
LPDIRECTDRAW2 lpDD2;
DDSCAPS      ddsCaps;
DWORD         dwTotal;
```

```
DWORD          dwFree;

ddres = lpDD->QueryInterface(IID_IDirectDraw2, &lpDD2);
if (FAILED(ddres))
{
    .
    .
    .
}

ddsCaps.dwCaps = DDSCAPS_TEXTURE;
ddres = lpDD2->GetAvailableVidMem(&ddsCaps, &dwTotal, &dwFree);
if (FAILED(ddres))
{
    .
    .
    .
}
```

- `CE,,,DD_OK,; Ž”,,,Ž,,,,ff ;;`
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_UNSUPPORTEDMODE

lpDDSurfaceDesc

f fff,Ž,DDSURFACEDESC “,,ffff

IDirectDraw2::RestoreDisplayMode

IDirectDraw2::SetDisplayMode

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

IDirectDraw2::GetFourCCCodes

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

DirectDraw

FourCC

㉔㉔

- ㉔,, DD_OK Ž”,,Ž,ff ,*,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lpNumCodes

*lpCodes,Ž,,,—,ffff ffff ,,,,,,,f f,Ž”,,,, ,,
lpNumCodes,—,’,fff,, , , ,lpCodes,“,*

lpCodes

*FourCCf f,“,—,ffff NULL,“,,,
lpNumCodes,, FourCCf f,ff f,,’,fff,, ff f,,*

IDirectDraw2::GetGDISurface

`HRESULT GetGDISurface(LPDIRECTDRAWSURFACE FAR * lplpGDIDDSurface);`

GDI

DirectDrawSurface

- ㉔,, DD_OK Ž”,,Ž,ff ,*,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_NOTFOUND

lplpGDIDDSurface

GDI,ffffff f fff,ffff f,,,DirectDrawSurfaceffffff,,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(LPWORD lpdwScanLine);`

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

lpdwScanLine
`ffffff` `Ž` `•` `ffff`

IDirectDraw2::GetVerticalBlankStatus
IDirectDraw2::WaitForVerticalBlank

IDirectDraw2::GetVerticalBlankStatus

`HRESULT GetVerticalBlankStatus(LPBOOL lpbIsInVB);`

⑦

- `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

fff ffff GUID ffff ffff
COM ¥ ¥
DirectDraw DirectDrawCreate
DDERR_ALREADYINITIALIZED IDirectDraw2::Initialize
DirectDraw CoCreateInstance
DDERR_NOTINITIALIZED

CoCreateInstance
CoCreateInstance

IDirectDraw2::Initialize
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**

```
HRESULT SetCooperativeLevel(HWND hWnd, DWORD dwFlags);
```

- `DDERR_EXCLUSIVEMODEALREADYSET`
`DDERR_HWNDALREADYSET`
`DDERR_HWNDSUBCLASSED`
`DDERR_INVALIDOBJECT`
`DDERR_INVALIDPARAMS`
`DDERR_OUTOFMEMORY`

ffff fff,Ž,,fffff¥ffff

$$\check{Z}, fff, ', \dots, \dots, \hat{}, \check{Z},$$

Mode Xfffffff f,ž—,‹‰

”“ f f f f f f f f f Ž, CTRL+ALT+DEL,“,%

”“fff,—<

”“f f, —Ž,‘ffffff f fff,‘,,%oo“, ,,,,Ž, GDI,—Ž,,,,,

ffff fff,' ,Windowsffff fff,,<",,,,Ž,

```
fffff fŽ DirectDraw,ffff ffffffff, %0,,,ffff,,,,,,,,
```

IDirectDraw2::Compact

DDSCL_FULLSCREEN

IDirectDraw2::SetCooperativeLevel

DDSCl EXCLUSIVE

DDSCL NORMAL

DDSCL_EXCLUSIVE

DDSCL_FULLSCREEN の

X

DDSCL ALLOWMODEX

DDSCL_FULLSCREEN

DDSCl EXCLUSIVE

DDSCAL ALLOWMODEX

DDSCl NORMAL

DDSCCL ALLOWMODEX

IDirectDraw2::EnumDisplayModes

X

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

Windows 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,€, ,
2 fffffff f,• ,,,, **IDirectDraw::SetDisplayMode**,€, ,
3
”“f f,%o ,,,,DDSCL_NORMALfff,<,**IDirectDraw::SetCooperative**
Level,€, ,

IDirectDraw2

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

IDirectDraw2::SetDisplayMode

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

¥

- €,,, 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,ffffffff 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 ,•,

DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_UNSUPPORTED
DDERR_WASSTILLDRAWING

dwFlags
'< ŠŠ,‘ ,,,,‘,,,Œ’,,
DDWAITVB_BLOCKBEGIN
'< ŠŠ,ŠŽŽ,,•,
DDWAITVB_BLOCKBEGINEVENT
'< ,Ž,,,,ffff fff ,,Œ ff f,,,,,
DDWAITVB_BLOCKEND
'< ŠŠ, ,, •Ž,ŠŽ,,,,•,,

hEvent
'< ,Ž,,,,,fff,,,ffff ffff
IDirectDraw2::GetVerticalBlankStatus IDirectDraw2::GetScanLine

IDirectDrawClipper

IDirectDrawClipper

IDirectDrawClipper

IDirectDrawClipper

fff” Initialize

ffffff fff GetClipList
IsClipListChanged
SetClipList
SetHWND

ffff GetHWND

IDirectDrawClipper COM
IUnknown Iunknown
の

AddRef
QueryInterface
Release

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
 fffffff fff,ffffff,,,,Ž—,‹€, “,,ffff

```
lpClipList
ffffffffff ffff,fff 0x%08X,RGNDATA “.,ffff
```

```
lpdwSize
ffffff fff,fff, ',• ,ffff
RGNDATA Q
```

```
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;
```

Win32SDK の ⑨⑨

IDirectDrawClipper::SetClipList

IDirectDrawClipper::GetHWnd

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

IDirectDrawClipper::SetHWND

DirectDrawClipper

¥

- Ē,,, DD_OK Ž”,,,Ž,ff ,*,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS

lphWnd

,,,**IDirectDrawClipper::SetHWND**ffff,,,
DirectDrawClipperffffff,Š~,,,,ffffff¥ffff,ffff
IDirectDrawClipper::SetHWND

IDirectDrawClipper::Initialize

HRESULT Initialize(LPDIRECTDRAW lpDD, DWORD dwFlags);

DirectDrawClipper

CoCreateInstance OLE

- Ē,,, DD_OK Ž”,,,Ž,ff ,*,
DDERR_ALREADYINITIALIZED
DDERR_INVALIDPARAMS

lpDD

DirectDrawffffff, DirectDraw “,ffff ,fff f,NULL,fff,,, “—
,,DirectDrawClipperffffff, ,,, (**DirectDrawCreateClipper**Š ,Ž,,,“,,
)

dwFlags

,fff f,Ē —Ž—,,, 0,fff,,,,,

の¥ ¥ (COM)

DirectDrawCreateClipper **IDirectDraw2::CreateClipper**

DirectDrawClipper

DDERR_ALREADYINITIALIZED**IDirectDrawClipper::Initialize** **CoCreateInstance***CoCreateInstanc* *DirectDrawClipper***IUnknown::AddRef** **IUnknown::QueryInterface****IUnknown::Release** **IDirectDraw2::CreateClipper****IDirectDrawClipper::IsClipListChanged**

HRESULT IsClipListChanged(BOOL FAR * lpbChanged);

¥ DirectDrawClipper

- Ē,,, DD_OK Ž”,,,Ž,ff ,*,

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
—RGNDATA “,fff, ,,,NULL ,,,DirectDrawClipper
,~,,,,ffffff fff,’ , ,,,,’,NULL,,, fffffff fff, ,,,

dwFlags
,fff f, —Ž—, 0,fff,,,,,,,,,
¥ 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;
```

IDirectDrawPalette::GetCaps

HRESULT GetCaps(LPDWORD lpdwCaps);

- **DD_OK**, **DDERR_INVALIDOBJECT**, **DDERR_INVALIDPARAMS**

lpdwCaps
DDPCAPS_4BIT
DDPCAPS_8BIT
DDPCAPS_8BITENTRIES
DDPCAPS_ALLOW256
DDPCAPS_PRIMARYSURFACE
DDPCAPS_PRIMARYSURFACELEFT
DDPCAPS_VSYNC

IDirectDrawPalette::GetEntries

HRESULT GetEntries(DWORD dwFlags, DWORD dwBase, DWORD dwNumEntries, LPPALETTEENTRY lpEntries);

DirectDrawPalette

- **DD_OK**, **DDERR_INVALIDOBJECT**, **DDERR_INVALIDPARAMS**, **DDERR_NOTPALETTIZED**

dwFlags
DDGDI_32BIT
dwBase
0
dwNumEntries
lpEntries, dwStartingEntry, dwCount, IDirectDrawPalette::SetEntries, lpEntries
DDPCAPS_8BITENTRIES

IDirectDrawPalette::Initialize

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

DirectDrawPalette

- **DDERR_ALREADYINITIALIZED**,•,

lpDD

DirectDraw fffffff,^-,DirectDraw “,ffff

dwFlags and *lpDDColorTable*

,,,fff f,Œ -Ž— 0,fff,,,,,

COM

∅∅

NULL

DirectDrawPalette

DDERR_ALREADYINITIALIZED

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

IUnknown::Release

IDirectDrawPalette::SetEntries

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

DirectDrawPalette

- Œ,,, DD_OK Ž”,,,Ž,ff ,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_NOPALETTEATTACHED

DDERR_NOTPALETTIZED

DDERR_UNSUPPORTED

dwFlags

,,,fff f,Œ -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
の
DirectDrawSurface
ののの

IDirectDrawSurface2

fff,Š,“,	Initialize
	IsLost
	Restore
f fff,ffff	AddAttachedSurface
	DeleteAttachedSurface
	EnumAttachedSurfaces
	GetAttachedSurface
ffff““	Blt
	BltBatch
	BltFast
ff f	GetColorKey
	SetColorKey
ffff fffffff	GetDC
	ReleaseDC
f fff fffffff	Flip
f fff fff	Lock
	PageLock
	PageUnlock
	Unlock
„‘	GetDDInterface
f f ff	AddOverlayDirtyRect
	EnumOverlayZOrders
	GetOverlayPosition
	SetOverlayPosition
	UpdateOverlay
	UpdateOverlayDisplay

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

IDirectDrawSurface2*IUnknown*

⑦

AddRef

QueryInterface

Release

COM

IUnknown**IDirectDrawSurface2::AddAttachedSurface**

```
HRESULT AddAttachedSurface(
    LPDIRECTDRAWSURFACE2 lpDDSAttachedSurface);
```

⑦

- Ⓔ,,, 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*

Z () 3D Z の の の () の の の Z の の の の の の **IDirectDrawSurface2::Flip** の の

IDirectDrawSurface2::DeleteAttachedSurface
IDirectDrawSurface2::EnumAttachedSurfaces **IDirectDrawSurface2::Flip**

IDirectDrawSurface2::AddOverlayDirtyRect

`HRESULT AddOverlayDirtyRect(LPRECT lpRect);`

IDirectDrawSurface2::UpdateOverlayDisplay

- Ǝ,,, DD_OK Ž”,,,Ž,ff ,*,
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_INVALIDSURFACETYPE
DDERR_UNSUPPORTED

lpRect
,,•—,,, **RECT** “,,ffff

IDirectDrawSurface2::UpdateOverlayDisplay

IDirectDrawSurface2::Blt

`HRESULT Blt(LPRECT lpDestRect, LPDIRECTDRAWSURFACE2 lpDDSrcSurface,
LPRECT lpSrcRect, DWORD dwFlags, LPDDBLTFX lpDDBltFx);`

- Ǝ,,, 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

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,“-%o,,,,,•, 0,•“—

DDBLT_ALPHADESTSURFACEOVERRIDE

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

DDBLT_ALPHAEDGEBLEND

ff f ,%Ž,%o‘,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,“-%o,,,,,•, 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 ,““ ĆE,—,, RGB ,,, DDBLTFX “,dwFillColor fff,Ž
—,,

DDBLT_DDFX

,,ffff“,”Ž—,,Ć‰,Ž’,,,,, DDBLTFX “,dwDDFX fff,Ž—,,

DDBLT_DDROPS

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

DDBLT_DEPTHFILL

““ Zffff f fff ,““ ĆE,—,, “”,,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
““Œ,,, 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

- Œ,,, 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,Œ -Ž— 0,fff,,,,,

IDirectDrawSurface2::BltFast

HRESULT BltFast(DWORD dwX, DWORD dwY,
LPDIRECTDRAWSURFACE2 lpDDSrcSurface, LPRECT lpSrcRect,
DWORD dwTrans);

- Œ,,, 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““ ,“Œ DirectDraw f fff,ffff
lpSrcRect
ffff““,„Œf fff ,Œ, ,,,%o%, •,’Œ, **RECT** “,ffff
dwTrans
““fff
DDBLTFAST_DESTCOLORKEY
““ ,ff f ,Ž—,„%oŒffff““,Ž’,, .
DDBLTFAST_NOCOLORKEY
“%o,,,’ ,ff ffff““,Ž’,,
DDBLTFAST_SRCOLORKEY
“Œ,ff f ,Ž—,„%oŒffff““,Ž’,,
DDBLTFAST_WAIT
ffff““,ff ,
DDERR_WASSTILLDRAWING•’,,,,•,,,‘, ,,,ffff““, ”,,,,,,,‘,ff
,” Ž‘ ,,,,–,

IDirectDrawSurface2::DeleteAttachedSurface

()

- ```
dwwFlags
 ,fff f,& -Ž—,, 0,fff,,,,,,,,,,,,,
lpDDSAttachedSurface
 ffff,, DirectDraw f fff,ffff
 ,fff f,NULL, ffff,,,,,,,,f fff,ffff,,
IDirectDrawSurface2::AddAttachedSurface
 DirectDraw
```

## IDirectDrawSurface2::Flip

## IDirectDrawSurface2::EnumAttachedSurfaces

- $\mathbb{E}_{,,,} \text{DD\_OK} \quad \check{Z}''_{,,,} \check{Z}_{,ff} \quad , \bullet ,$

---

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

*lpContext*  
f ffffŠ ,“,,, ffff fff,’€, “,,ffff  
*lpEnumSurfacesCallback*  
ffff,,,,,f fff–,€, ,, EnumSurfacesCallback Š ,ffff

## IDirectDrawSurface2::EnumOverlayZOrders

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

Ⓞ

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

*dwFlags*  
^‰,fff,,,,,,,,,,’,Ž,  
**DDENUMOVERLAYZ\_BACKTOFRONT**  
fff,,ffff,€,f f ff,—€, ,  
**DDENUMOVERLAYZ\_FRONTTOBACK**  
ffff,,fff,€,f f ff,—€, ,

*lpContext*  
Šf f ff f fff,f ffffŠ ,“,,,f f’<ffffff,,ffff  
*lpfnCallback*  
,,f fff ,f f ff,,f fff–,€, ,,,, fnCallback Š ,ffff

## IDirectDrawSurface2::Flip

HRESULT Flip(  
LPDIRECTDRAWSURFACE2 lpDDSurfaceTargetOverride,  
DWORD dwFlags);

DDSCAPS\_BACKBUFFER

- €,,, 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',Œ,,ffff,,  
 ,, ,ffff,, ,,fff f,f f fff,,,,Ž—,,

*dwFlags*

**DDEFLIP\_WAIT**

^", ffffff f ffff, ‘“ ,,,ffffff, ”,,,,  
 DDERR\_WASSTILLDRAWING  
 ff ,,,,\*, ffffff,‘,,, ,,, ,fff,fff,,  
 HAL,,DDERR\_WASSTILLDRAWING  
 ff ,Ž,Ž,,**IDirectDrawSurface2::Flip**,ffffff,,,,‘,,  
 ,, **IDirectDrawSurface2::Flip**,ffffff —, Œ,,  
 DDERR\_SURFACEBUSY,,,\*,ff ,\*,,,—,,

DDSCAPS\_FLIP

DDSCAPS\_FRONTBUFFER

Ō

Ō

**IDirectDrawSurface2::Flip**

*lpDDSurfaceTargetOverride*

NULL

**IDirectDrawSurface2::Flip**

**IDirectDrawSurface2::GetFlipStatus**

## **IDirectDrawSurface2::GetAttachedSurface**

HRESULT GetAttachedSurface(LPDDSCAPS lpDDSCaps,  
 LPDIRECTDRAWSURFACE2 FAR \* lpIAttachedSurface);

- Œ,, DD\_OK Ž”,,Ž,ff ,\*,

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

*lpDDSCaps*

f fff,f ffff—,Ž, DDSCAPS “,ffff

---

*lpDDAttachedSurface*

**EnumSurfacesCallback** *f* *ffffŠ* , *lpDDSurface*  
*fff f,Ž* , , *Œ* , *DirectDraw f fff,ffff,,* , , *lpDDSCaps*  
*fff f,Ž* , , *Œ* , , *—,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*

*^%o,fff,,,,,,,,,Ž,*

**DDGBS\_CANBLT**

*,,f fff,Š,,ffff“,,,,Ž* , , , , , , , , *ffff“„Š—,,* *DD\_OK* , , ,

**DDGBS\_ISBLTDONE**

*ffff“„* , , , , , , , , , , *,,f fff, Œ,ffff“„Š—,,,,,DD\_OK* , , ,

## **IDirectDrawSurface2::GetCaps**

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

∅

- Œ,,* *DD\_OK Ž* , , , *Ž,ff* , , ,

**DDERR\_INVALIDOBJECT**

**DDERR\_INVALIDPARAMS**

*lpDDSCaps*

*f fff,f ffff*—,“,DDSCAPS “,ffff

## IDirectDrawSurface2::GetClipper

HRESULT GetClipper(LPDIRECTDRAWCLIPPER FAR \* lpDDClipper);

⑦

DirectDrawClipper

- ⑆,,, 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 ⑦

- ⑆,,, 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 ,Œ ,’Š”,DDCOLORKEY  
“,ffff

**IDirectDrawSurface2::SetColorKey**

## **IDirectDrawSurface2::GetDC**

HRESULT GetDC(HDC FAR \* lphDC);

のの

GDI

- Œ,,, 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 \*lpplDD);

DirectDraw

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

*lpplDD*

DirectDraw ffff,\*,ffff,ffff Œ, ,,Ž”,, ,,’–Œ,,,

**IDirectDraw2**

**IDirectDraw**

## IDirectDrawSurface2::GetFlipStatus

HRESULT GetFlipStatus(DWORD dwFlags);

- DD\_OK f fff,ffffff —, ,,,,,,  
DDERR\_WASSTILLDRAWING ,‘,—,Ž”,,Ž,ff ,\*,  
DDERR\_INVALIDOBJECT  
DDERR\_INVALIDPARAMS  
DDERR\_INVALIDSURFACETYPE  
DDERR\_SURFACEBUSY  
DDERR\_SURFACELOST  
DDERR\_UNSUPPORTED

*dwFlags*

^%o,fff,,,,,Ž’,,

DDGFS\_CANFLIP

,,f fff,,,,ffffff,,,,,—, , ffffff,Š—,, DD\_OK,\*,

DDGFS\_ISFLIPDONE

ffffff, ,,—, , ,f fff, Ą,ffffff,Š—,,, ,DD\_OK,\*,

IDirectDrawSurface2::Flip

## IDirectDrawSurface2::GetOverlayPosition

HRESULT GetOverlayPosition(LPLONG lplX, LPLONG lplY);

○

DDSCAPS\_OVERLAY

○○

- Ą,, DD\_OK Ž”,,Ž,ff ,\*,  
DDERR\_GENERIC  
DDERR\_INVALIDOBJECT  
DDERR\_INVALIDPARAMS  
DDERR\_INVALIDPOSITION  
DDERR\_NOOVERLAYDEST  
DDERR\_NOTAOVERLAYSURFACE  
DDERR\_OVERLAYNOTVISIBLE  
DDERR\_SURFACELOST

*lplX and lplY*

xffffff \*,,yffffff \*,ffff

IDirectDrawSurface2::SetOverlayPosition

IDirectDrawSurface2::UpdateOverlay

## IDirectDrawSurface2::GetPalette

```
HRESULT GetPalette(LPDIRECTDRAWPALETTE FAR * lpDDPalette);
```

## DirectDrawPalette

- DD\_OK
- DDERR\_GENERIC  
DDERR\_INVALIDOBJECT  
DDERR\_INVALIDPARAMS  
DDERR\_NOEXCLUSIVEMODE  
DDERR\_NOPALETTEATTACHED  
DDERR\_SURFACELOST  
DDERR\_UNSUPPORTED

*lplpDDPalette*

```

Ž“,,f fff,Š~,,,, DirectDrawPalette “,fffff,ffff
,,f fff,Š~,,,, DirectDrawPalette “,,, NULL ,fff,,
,,fff f, f fff,ffffff f fff, ,,,ffffff f fff,fff ffff, ,ff
fff f fff,8 bppf f,, ffff ffff,,ffff,Ž,

```

## IDirectDrawSurface2::SetPalette

## IDirectDrawSurface2::GetPixelFormat

```
HRESULT GetPixelFormat(LPDDPIXELFORMAT lpDDPixelFormat);
```

 $\mathcal{O}$ 

- `E,, DD_OK` `Ž”,,Ž,ff` `;`  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_INVALIDSURFACETYPE**

*lpDDPixelFormat*

Ⓒ ffff, •,f fff,ff ff f ff fff, ,< ,Ž“,DDPIXELFOR  
MAT “,,ffff

## IDirectDrawSurface2::GetSurfaceDesc

```
HRESULT GetSurfaceDesc(LPDDSURFACEDESC lpDDSurfaceDesc);
```

②②

## DDSURFACEDESC

- `DD_OK` `DDERR_INVALIDOBJECT` `DDERR_INVALIDPARAMS`

```

lpDDSurfaceDesc
 f fff, 'Ž“, DDSURFACEDESC ‘, ffff
 DDSURFACEDESC

```

## IDirectDrawSurface2::Initialize

```

HRESULT Initialize(LPDIRECTDRAW lpDD,
 LPDDSURFACEDESC lpDDSurfaceDesc);

```

DirectDrawSurface

- **DDERR\_ALREADYINITIALIZED**,•,

```

lpDD
 DirectDraw fffffff,•, DirectDraw ‘, ffff
lpDDSurfaceDesc
 f fff,,,,, •, ', DDSURFACEDESC ‘, ffff
 の COM
 DirectDrawSurface
 DDERR_ALREADYINITIALIZED

```

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

## IDirectDrawSurface2::IsLost

```

HRESULT IsLost();

```

DirectDrawSurface

- Ž,,,,,, DD\_OK ,,,,Ž,ff ,•,  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_SURFACELOST**

```

 ののの
DirectDrawSurface の
DDERR_SURFACELOST
 ののの

```

**IDirectDrawSurface2::Restore**

## IDirectDrawSurface2::Lock

```

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

```

の

- **DD\_OK** **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*, •, , %” , , , *ffff*, <  
“, *fff* *ffff* *ffff*, **IDirectDrawSurface2::Lock** “, , , , *fff*, ,  
, *fff*, *fff*, *f* *fff* , • , , , *ffff*, FIFO , ^, <, , ,

**DDLOCK\_READONLY**

*fff*, *f* *fff*, “, , —, , , , *Ž*, *fff*

**DDLOCK\_SURFACEMEMORYPTR**

*Ž*, , < , “, —, *fff* *ffff*, •, , , , , , •, *fff*  
< , *Ž*, , , , “ , *f* *fff*, *ffff*, •, , *ffff*, , , *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**

の

**DDERR\_SURFACEBUSY**

**DDERR\_LOCKEDSURFACES**





の

- `DD_OK` `DDERR_CANTPAGEUNLOCK`  
`DDERR_INVALIDOBJECT`  
`DDERR_INVALIDPARAMS`  
`DDERR_NOTPAGELOCKED`  
`DDERR_SURFACELOST`

*dwFlags*

`DDI_WAITONRELEASE` `DDI_WAITONRELEASE`

`IDirectDrawSurface2::PageLock`

`IDirectDrawSurface2::PageUnlock`

0

の

のののの

`DD_OK`

`IDirectDraw2`

`IDirectDraw`

`IDirectDrawSurface2::PageLock`

## `IDirectDrawSurface2::ReleaseDC`

`HRESULT ReleaseDC(HDC hDC);`

`IDirectDrawSurface2::GetDC`

- `DD_OK` `DDERR_GENERIC`  
`DDERR_INVALIDOBJECT`  
`DDERR_INVALIDPARAMS`  
`DDERR_SURFACELOST`  
`DDERR_UNSUPPORTED`

*hDC*

`IDirectDrawSurface2::GetDC` `DDI_WAITONRELEASE` `DDI_WAITONRELEASE`

`IDirectDrawSurface2::GetDC`

の

`IDirectDrawSurface2::GetDC`

## `IDirectDrawSurface2::Restore`

`HRESULT Restore();`

- `DDERR_GENERIC`
- `DDERR_IMPLICITLYCREATED`
- `DDERR_INCOMPATIBLEPRIMARY`
- `DDERR_INVALIDOBJECT`
- `DDERR_INVALIDPARAMS`
- `DDERR_NOEXCLUSIVEMODE`
- `DDERR_OUTOFMEMORY`
- `DDERR_UNSUPPORTED`
- `DDERR_WRONGMODE`

DirectDrawSurface の  
DDERR\_SURFACELOST  
**IDirectDrawSurface2::Restore**  
DirectDrawSurface

**IDirectDrawSurface2::Restore**  
**IDirectDrawSurface2::AddAttachedSurface**

**IDirectDrawSurface2::IsLost**  
**IDirectDrawSurface2::AddAttachedSurface**

```
HRESULT SetClipper(LPDIRECTDRAWCLIPPER lpDDClipper);
```

- `DDERR_INVALIDOBJECT`  
`DDERR_INVALIDPARAMS`  
`DDERR_INVALIDSURFACETYPE`  
`DDERR_NOCLIPPERATTACHED`

```
lpDDClipper
 DirectDrawSurface ffffff,ffff,, DirectDrawClipper
 ffffff,*,,, DirectDrawClipper “,ffff
 ,,,,fff f,ff,,, ☺ , DirectDrawClipper ffffff,ffff,,,,^-,
```

---

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

- Ǝ,,, DD\_OK Ž”,,,Ž,ff ,\*,  
DDERR\_GENERIC  
DDERR\_INVALIDOBJECT  
DDERR\_INVALIDPARAMS  
DDERR\_INVALIDSURFACETYPE  
DDERR\_NOOVERLAYHW  
DDERR\_NOTAOVERLAYSURFACE  
DDERR\_SURFACELOST  
DDERR\_UNSUPPORTED  
DDERR\_WASSTILLDRAWING

dwFlags  
,,,ff f ,—<,,,Ž’,,  
DDCKEY\_COLORSPACE  
“,ff ff f,Š,,, fff,, “,’^ff f ,Š,,,,,fff,,,,,  
DDCKEY\_DESTBLT  
ffff““ ff f ,,Ž—  
,,,ff f ,,,,“ ff ff f, “,Ž’,,, fff,,  
DDCKEY\_DESTOVERLAY  
f f ff —,“ ff f ,,Ž—  
,,,ff f ,,,,“ ff ff f, “,Ž’,,, fff,,  
DDCKEY\_SRCBLT  
ffff““Ǝff f ,,Ž—  
,,,ff f ,,,,“ ff ff f, “,Ž’,,, fff,,  
DDCKEY\_SRCOVERLAY

*f f ff —,“Œff 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);

○

- Œ,,, 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(LPDIRECTDRAWPALETTE lpDDPalette);

DirectDrawPalette

○○

- Œ,,, 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

- Ē,,, 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,^,,“ <Ē,,,•  
, **IDirectDrawSurface2::Lock**(Ē, ,,%”,,, ,ffff, **IDirectDrawSurface**  
**2::Lock**,, **IDirectDrawSurface2::Unlock**(Ē, ,,Ē,•,,  
**IDirectDrawSurface2::Lock**

## **IDirectDrawSurface2::UpdateOverlay**

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

の

の

DDSCAPS\_OVERLAY

- Ē,,, DD\_OK Ž”,,,Ž,ff ,•,  
**DDERR\_GENERIC**  
**DDERR\_HEIGHTALIGN**  
**DDERR\_INVALIDOBJECT**  
**DDERR\_INVALIDPARAMS**  
**DDERR\_INVALIDRECT**  
**DDERR\_INVALIDSURFACETYPE**  
**DDERR\_NOSTRETCHHW**  
**DDERR\_NOTAOVERLAYSURFACE**

```

lpSrcRect
 ““Ⓔf fff,f f ff,Ž,,• ,•,,,—^, ,,’^,, RECT “,fffff
lpDDDestSurface
 f f ff,,,,, DirectDraw f fff,fffff
lpDestRect
 f f ff, ,,”“ f fff, ,• ,•,,,—^, ,,’^,, RECT “,fffff
dwFlags
 DDOVER_ADDDIRTYRECT
 ffff fff,f f ff,,,f fff, ,,,,,,,—^,%0,,
 DDOVER_ALPHADEST
 ffff ff fff,,ffff •,,,ffff ffff f fff, ,f f ff,”“ ffff
 ffff,,,f fff,ffff,„„Ž—,,
 DDOVER_ALPHADESTCONSTOVERRIDE
 ,,f f ff,”“ ffff ffff,,,DDOVERLAYFX “,
 dwAlphaDestConst fff,Ž—,,
 DDOVER_ALPHADESTNEG
 ffff’,%0,,,”“ f fff,—%0,,,,,•, 0,•—
 DDOVER_ALPHADESTSURFACEOVERRIDE
 ,,f f ff,”“ ffff ffff,,,DDOVERLAYFX “,lpDDSAphaDest
 fff,Ž—,,
 DDOVER_ALPHAEDGEBLEND
 ff f ^, ,,%‘,fff,ffff ffff,,,DDOVERLAYFX “,
 dwAlphaEdgeBlend fff,Ž—,,
 DDOVER_ALPHASRC
 ffff ff fff,,ffff •,,,ffff ffff f fff, ,f f ff,”“Ⓔffff
 ffff,,,f fff,ffff,„„Ž—,,
 DDOVER_ALPHASRCCONSTOVERRIDE
 ,,f f ff,”“Ⓔffff ffff,,,DDOVERLAYFX
 “,dwAlphaSrcConst fff,Ž—,,
 DDOVER_ALPHASRCNEG
 ffff’,%0,,,”Ⓔf fff,—%0,,,,,•, 0,•—
 DDOVER_ALPHASRCSURFACEOVERRIDE
 ,,f f ff,ffff ffffⒺ,,,DDOVERLAYFX “,lpDDSAphaSrc
 fff,Ž—,,
 DDOVER_DDFX
 “Ž,f f ffⒺ%0,’^,,,,, f f ffFXfff,Ž—,,
 DDOVER_HIDE
 ,,f f ff,ff,,
 DDOVER_KEYDEST

```

```

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

lpDDOverlayFx
DDOVERLAYFX “,Ž ,,,

```

## IDirectDrawSurface2::UpdateOverlayDisplay

HRESULT UpdateOverlayDisplay(DWORD dwFlags);

```

 ののの
 の

```

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

```

dwFlags
 —,, fff ^%0,fff,,,,,,’,Ž,
DDOVER_REFRESHDIRTYRECTS
 ,,“ ,,,‘ ,,—^,, ,,,Œ,fff,Ž—
 ,, fffffff, ,,, ,,,Œfff,fff,,
DDOVER_REFRESHALL
 ,,,Œ,fff,—Ž,, f f ff fffffff,Š‘, ,,, ,,,Œfff,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

**lpDDSSrc**

ffff“Ė,’<,, DirectDrawSurfacefffff,ffff

**lprSrc**

ffff“Ė,Ė,’<,, RECT “,ffff

**dwFlags**

ffff Ėfff

**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“Ėf fff,ffff ffff,, DDBLTFX “,  
dwAlphaSrcConst fff,Ž—,,

**DDBLT\_ALPHASRCNEG**

ffff’,%o,,“Ėf fff,“-%o„„„,•, 0,•“—

**DDBLT\_ALPHASRCSURFACEOVERRIDE**

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

**DDBLT\_ASYNC**

FIFO f ffff’,Ž , , ,fff“„”Š,Ž , , ,FIFO f ffff,Ė—  
^„„„ Ė, ,Ž”,,

**DDBLT\_COLORFILL**

ffff“ f fff ,ffff“ ĖĖ,—, RGB „„ DDBLTFX “,  
dwFillColor fff,Ž—,,

**DDBLT\_DDFX**

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

**DDBLT\_DDROPS**

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

```

 dwDDROPS fff,Ž—,,
DDBLT_KEYDEST
 ffff“ f fff,Œ•,,ff f ,Ž—,,
DDBLT_KEYDESTOVERRIDE
 ffff“ f fff,ff f ,, DDBLTFX “,dckDestColorkey fff,Ž
 —,,
DDBLT_KEYSRC
 ffff“Œf fff,Œ•,,ff f ,Ž—,,
DDBLT_KEYSRCOVERRIDE
 •Œf 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_ZBUFFER
 ffff“Œ,,,ffff“ f fff,ffff,,Zffff,Ž—,, Zffff ffff“ , ,
 DDBLTFX “, dwZBufferOpCode fff, Zffff fff f,,
DDBLT_ZBUFFERDESTCONSTOVERRIDE
 DDBLTFX “,dwZDestConst,“ Zffff,Ž—
 ,, Zffff ffff“Ž , , “ “,dwZBufferOpCodefff,“ Zffff fff
 f,,
DDBLT_ZBUFFERDESTOVERRIDE
 DDBLTFX “,lpDDSZBufferDest,“ Zffff,Ž—
 ,, Zffff ffff“Ž , , “ “,dwZBufferOpCode
 fff,“ Zffff fff f,,
DDBLT_ZBUFFERSRCCONSTOVERRIDE
 DDBLTFX “, dwZSrcConst,“ŒZffff,Ž—
 ,, Zffff ffff“Ž , , “ “,dwZBufferOpCode
 fff,“ŒZffff fff f,,
DDBLT_ZBUFFERSRCOVERRIDE
 DDBLTFX “, lpDDSZBufferSrc,“ŒZffff,Ž—
 ,, Zffff ffff“Ž , , “ “, dwZBufferOpCode
 fff,“ŒZffff fff f,,
lpDDBlTFx
 ’%o,,,ffff“Œ%o,Ž’,, DDBLTFX “,ffff

```

## DDBLTFX

```

typedef struct _DDBLTFX{
 DWORD dwSize;
 DWORD dwDDFX;
 DWORD dwROP;
 DWORD dwDDROP;

```

---

```

 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 lpDDSAAlphaDest;
};
 DWORD dwAlphaSrcConstBitDepth;
union
{
 DWORD dwAlphaSrcConst;
 LPDIRECTDRAWSURFACE lpDDSAAlphaSrc;
};
union
{
 DWORD dwFillColor;
 DWORD dwFillDepth;
 LPDIRECTDRAWSURFACE lpDDSPattern;
};
DDCOLORKEY ddckDestColorkey;
DDCOLORKEY ddckSrcColorkey;
} DDBLTFX, FAR* LPDDBLTFX;

```

## IDirectDrawSurface2::Blt

### IDirectDrawSurface2::BltBatch

#### DDBLTBATCH の

#### dwSize

“ ‘ , , , , f f f , “ , Ž , , , ‘ , ’ , , , , , , , , , ,

#### dwDDFX

Œ% —, f f f , Ž ’ , f f f

#### DDBLTFX\_ARITHSTRETCHY

, , f f f f “ , , , , Ž , % , , % Ž ’ , Ž —, ,

**DDBLTFX\_MIRRORLEFTRIGHT**

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

**DDBLTFX\_MIRRORUPDOWN**

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

**DDBLTFX\_NOTEARING**

,ffff“ ,,,,ffffff,”,,,, —,,

**DDBLTFX\_ROTATE180**

,ffff“ ,Š,f fff,ŽŒ%o,,180“%o“,,,

**DDBLTFX\_ROTATE270**

,ffff“ ,Š,f fff,ŽŒ%o,,270“%o“,,,

**DDBLTFX\_ROTATE90**

,ffff“ ,Š,f fff,ŽŒ%o,,90“%o“,,,

**DDBLTFX\_ZBUFFERBASEDEST**

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

**DDBLTFX\_ZBUFFERRANGE**

,ffff“ ,Š,ffff“Œf fff,• ,fff,ŒŠ,Ž’,,”^,,,dwZBufferLow,,, dwZBufferHigh fff,Ž—,,

**dwROP**

Win32 fff —

**dwDDROP**

DirectDraw fff —

**dwRotationAngle**

ffff“ ,%o“Š“

**dwZBufferOpCode**

Zffff”Š

**dwZBufferLow**

Zffff, ’ŒŠ

**dwZBufferHigh**

Zffff, ŒŠ

**dwZBufferBaseDest**

Zffff,ffff“ Š”

**dwZDestConstBitDepth**

ffff“ ,’ ,fff “

**dwZDestConst**

ffff“ Zffff,,,Ž—,,,’

**lpDDSZBufferDest**

ffff“ Zffff,,,Ž—,,,f fff

**dwZSrcConstBitDepth**

ffff“Œ,’ ,fff “

**dwZSrcConst**

ffff“ŒZffff,,,Ž—,,,’

### lpDDSZBufferSrc

ffff““ŒZffff,,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““Œffff’ ,fff “

### dwAlphaSrcConst

ffff““Œffff ffff,,Ž,,,’

### lpDDSAlphaSrc

ffff““Œffff ffff,,Ž—,,f fff

### dwFillColor

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

### dwFillDepth

Zffff, “

### lpDDSPattern

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

### ddckDestColorkey

— ,,ffff““ ff f

### ddckSrcColorkey

— ,,ffff““Œff f

## DDCAPS

```
typedef struct _DDCAPS{
 DWORD dwSize;
 DWORD dwCaps;
 DWORD dwCaps2;
 DWORD dwCKeyCaps;
 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,” ,’,%o” ,,,,,  
•,

**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\_CANBLTSYSMEM**

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

**DDCAPS\_CANCLIP**

ffffff f ffff,ffff““,ffffff,,,,\*,

**DDCAPS\_CANCLIPSTRETCHED**



```

 ffffff f ffff, 'ffff""',ffffff,,,,*,
DDCAPS_COLORKEY
 f f ff,,ffff""',,,,,ff f ,,,,,ŒŽ,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 ,ffffff 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
 ,,, Œ,f f ff,',,,IDirectDrawSurface2::UpdateOverlayZOrderf
 fff,Ž—,ff f,,

dwCaps2
 ,,ffff""—, •,fff
DDCAPS2_CERTIFIED
 ffffff f ffff,• ,,,,,*,
DDCAPS2_NO2DDURING3DSCENE
 IDirect3DDevice::BeginSceneffffŒ, ,, IDirect3DDevice::EndScene
 ffffŒ, ,,,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,
 ,
DDKEYCAPS_SRCOVERLAYCLRSPACEYUV
 YUV f f ff f fff,ffff““Œff f ,,,ff ff f,ff f,,
DDKEYCAPS_SRCOVERLAYONEACTIVE
 %Ž ‘,f f ff f fff, ,,,,—Œ,ffff““Œff f ’,ff f,,
DDKEYCAPS_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_BLTMIRRORLEFTRIGHT
 ffff““ ‘, %o“,,,,ff f,,
DDFXCAPS_BLTMIRRORUPDOWN
 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',ffffff,, ffff"" ,Š,Š', f fff, ,%oŽ —,Ž—  
 ,, ,Ž ' ,%o,, ,,,  
**DDFXCAPS\_OVERLAYARITHSTRETCHYN**  
 ffff,, ,,,Š',ffffff,, 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“Œff 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,, ,Œ,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,, ,Œ,fff,,,,,,
ffff““ ,,,,Ž—,,
```

**DDFXALPHACAPS\_OVERLAYALPHAEDGEBLEND**

```
ffff“Œff 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,, ,Œ,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,, ,Œ,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 fffffff,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, ,%Š," ,3DŠ<,ffff ff ,ŽŒ,,,,,,\*,  
,,,%“(ff ff f),,Ž‘,“ ,,,,,,,

**DDSVCAPS\_SPLIT**

•Šfff f,,ffff ff ,ŽŒ,,,,,,\*,

**dwAlphaBlitConstBitDepths**

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

**dwAlphaBlitPixelBitDepths**

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

**dwAlphaBlitSurfaceBitDepths**

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““ ŒŒ, •,’^

**dwAlignSizeDest**

ffff““ ŒŒ,‘,,,^

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

```

## DDCOLORKEY

```

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

```

⑦

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

⑦

```

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;
 LPDIRECTDRAWSURFACE lpDDAlphaDest;
};
 DWORD dwAlphaSrcConstBitDepth;
union
{
 DWORD dwAlphaSrcConst;
 LPDIRECTDRAWSURFACE lpDDAlphaSrc;
};
 DDCOLORKEY dckDestColorkey;
 DDCOLORKEY dckSrcColorkey;

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

```

### IDirectDrawSurface2::UpdateOverlay

#### dwSize

“‘,, ,fff, “Ž,,‘,’ ,,,,,,,,,,”

#### dwAlphaEdgeBlendBitDepth

ffff fff ffffffffŒ ,Ž’,,,,Ž—,,,fff “

#### dwAlphaEdgeBlend

ffff fff ffffffff,ffff,,Ž,,,’

#### dwReserved

—,Ž—,,,—,,,,,

#### dwAlphaDestConstBitDepth

ffff““ ,ffff’ ,Ž’,,,,Ž—,,,fff “

#### dwAlphaDestConst

ffff““ ,ffff ffff,,Ž,,,’

#### lpDDAlphaDest

ffff““ ,ffff ffff,,Ž,,,f fff,ffff

#### dwAlphaSrcConstBitDepth

ffff“Œ,ffff’ ,Ž’,,,,Ž—,,,fff “

#### dwAlphaSrcConst

ffff“Œ,ffff ffff,,Ž,,,’

#### lpDDAlphaSrc

ffff“Œ,ffff ffff,,Ž,,,f fff,ffff

#### dckDestColorkey

— ,ffff““ ff f

#### dckSrcColorkey

— ,ffff“Œff f

### dwDDFX

*f f ff EXfff*

### DDOVERFX\_ARITHSTRETCHY

*„Š‘,,, ,Ž,%o,,%oŽŠ‘,Ž—,,*

### DDOVERFX\_MIRRORLEFTRIGHT

*%o”“ , ,*

### DDOVERFX\_MIRRORUPDOWN

*%o”“ , ,*

### dwFlags

*„fff,Œ 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;
```

の IDirectDrawSurface2::GetPixelFormat  
のの

DirectDrawSurface

**dwSize**

“‘,, ,fff, “Ž,,‘,’ ,,,,,,,,,,

**dwFlags**

ffff Ćfff

**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,—Ć,,

**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,—Ć,,

**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,—Ć,,

**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

**dwRBitMask**

*fff,fff*

**dwYBitMask**

*,fff,fff*

**dwGBitMask**

*—fff,fff*

**dwUBitMask**

*,fff,fff*

**dwBBitMask**

*fff,fff*

**dwVBitMask**

*,fff,fff*

**dwRGBAAlphaBitMask**

*ffff ffff,fff*

**dwYUVAlphaBitMask**

*ffff ffff,fff*

## DDSCAPS

```
typedef struct _DDSCAPS{
 DWORD dwCaps;
} DDSCAPS,FAR* LPDDSCAPS;
```

DirectDrawSurface

の

DirectDraw

DDCAPS の

**dwCaps**

*f fff,”—,•, ~%o,fff,,,,,,,,,,,,,• ,‘ ,Ž,*

**DDSCAPS\_3D**

*,,%,^ŒŠ ,,,,ff f,,,,, ffff fff,‘,,, DDSCAPS\_3DDEVICE  
fff,Ž,,,,,*

**DDSCAPS\_3DDEVICE**

*,,f fff,,,•Œ—,—%o”,,,,•,  
ffff fff,ffff,“ ,f f,,Š,“,,,,fff fŠ,f fff,Ž,,,, “,f f,,,•Œ,,,,  
,,,,,• ,,,, ,fff,Ž—,,,,,  
,,,fff,f f,,,fff,,,,, f fff,,f f,,Š,“,,,,*

**DDSCAPS\_ALLOCONLOAD**

*f fff,IDirect3Dtexture::Loadffff,,,f f,,,, f fff,fff,Š,“,,,,,  
Ž’*

**DDSCAPS\_ALPHA**

*,,f fff,ffff •,Š,,,,,•,  
ffff ff fff,,,f fff,,,ffff •,,,ffff f f RGBA ,,,, YUVA  
,,, , ,,,,ffff •,,,Š,,,,,Œ’,,,,,Ž“,,,,,,*

**DDSCAPS\_BACKBUFFER**

*,,f fff,f fff ffffff “,fff ffff,,,,,•,  
^”,,,fff, DDSCAPS\_FLIPfff,Ž—  
,,,IDirectDraw2::CreateSurfaceffff,,, ’,,, DDSCAPS\_FRONTBU*

```

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 fffffff 'Œ, ,,,ffff ffff
,,,,fff ffff ,,,—•,,ffff,,,,,, ,
IDirectDrawSurface2::AddAttachedSurface,Ž ,,, DirectDraw
,ffffff 'Œ",, ,f fff,ff f,,

```

**DDSCAPS\_COMPLEX**

```

• f fff,< ,,,,,,•, • f fff,^ ,f fff, ,,,,
'%of fff,f f f fff,ffff,, • "Œ,f f,"Š,,,,,,,"Š,,,,,,

```

**DDSCAPS\_FLIP**

```

,,f fff,f fffffff 'Œ",,,,,,•,,"—
, IDirectDraw2::CreateSurfaceffff,",,,, ,,,ffff ffff,,,,,^ ,fff
ffff, ,, DirectDraw
,ffff ffff,DDSCAPS_FRONTBUFFERfff,,ffff ffff,—
,,,f fff, DDSCAPS_BACKBUFFERfff,fff,,
,fff,Ž',, DDSURFACEDESC 'Œ,
dwBackBufferCountfff, ,,,,fff,,,,,,
IDirectDraw2::CreateSurfaceffff,Ž—
,, ' ,f fff, ,Ž ,DDSCAPS_COMPLEXfff,fff,,,,,,

```

**DDSCAPS\_FRONTBUFFER**

```

,,f fff,f fff fffffff 'Œ,ffff ffff,,,,,•,
,,fff,^", DDSCAPS_FLIP "—
,fff,,,, IDirectDraw2::CreateSurfaceffff,,,,fff,,
IDirectDraw2::CreateSurfaceffff,Ž—,, "—Œffff ffff, ,,
,f fff, DDSCAPS_FLIPfff,Ž,, fffffff '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 fX) f fff,,,,,•,

```

**DDSCAPS\_OFFSCREENPLAIN**

```

,,f fff,f f ff fffff Žffff 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,Œ 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, -,Œ,,,,,“—,*,
 ,,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_VIDEOMEMORY
 ,,f fff,fffff fff,‘ ,,,,*,
DDSCAPS_VISIBLE
 ,,f fff,‘, ,, *, ‘ ,%—,”%,,,,,,,*, ,,fffff f fff,fff,,
DDSCAPS_WRITEONLY
 f fff, ,, —,,,,,,,*, f fff,,,“, ,ffff,~”•Œ ,, —
 Š, ,%” ,, ,,f fff,,,“, ,Œ%,“—,• ,,,,
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

ffffff Ćfff ~%,fff,,,,,,,,,• ,‘ ,Ž, “,,,fff,—Ć,,,Ž’,,

#### DDSD\_ALL

“—,,,,,,,,fff,—Ć,,,,,,,,•,

#### DDSD\_ALPHABITDEPTH

dwAlphaBitDepth fff,—Ć,,,,,,,,•,

#### DDSD\_BACKBUFFERCOUNT

dwBackBufferCount fff,—Ć,,,,,,,,•,

#### DDSD\_CAPS

ddsCaps fff,—Ć,,,,,,,,•,

#### DDSD\_CKDESTBLT

ddckCKDestBlt fff,—Ć,,,,,,,,•,

#### DDSD\_CKDESTOVERLAY

ddckCKDestOverlay fff,—Ć,,,,,,,,•,

#### DDSD\_CKSRCLT

ddckCKSrcBlt fff,—Ć,,,,,,,,•,

#### DDSD\_CKSRCOVERLAY

ddckCKSrcOverlay fff,—Ć,,,,,,,,•,

#### DDSD\_HEIGHT

dwHeight fff,—Ć,,,,,,,,•,

#### DDSD\_MIPMAPCOUNT

dwMipMapCount fff,—Ć,,,,,,,,•,

#### DDSD\_PITCH

lPitch fff,—Ć,,,,,,,,•,

#### DDSD\_PIXELFORMAT

ddpfPixelFormat fff,—Ć,,,,,,,,•,

#### DDSD\_REFRESHRATE

dwRefreshRate fff,—Ć,,,,,,,,•,

---

**DDSD\_WIDTH**

**dwWidth** *fff,—Œ,,,,,•,*

**DDSD\_ZBUFFERBITDEPTH**

**dwZBufferBitDepth** *fff,—Œ,,,,,•,*

**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*

**ddckCKDestBlt**

*ffff“ ,Ž—,,ff f*

**ddckCKSrcOverlay**

*f f ffŒ,Ž—,,ff f*

**ddckCKSrcBlt**

*ffff“Œ,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,ffff,,,,,,, ,f fff,ffff,“,”,,

**DDERR\_CANTPAGELOCK**

f fff,f f fff,Ž”,,

f f fff,ffffff fff f fff,,,ffff f,,,ffffff f fff, ,“ ,,,

**DDERR\_CANTPAGEUNLOCK**

f fff,f f fff%o ,Ž”,,

f f fff%o ,ffffff fff f fff,,,ffff f,,,ffffff f fff, ,“ ,,,

**DDERR\_CLIPPERISUSINGHWND**

ffffff ffff,,,fff ,,,,DirectDrawClipper

ffffff,ffffff fff,ffff,,,,,

**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,ffff,,,,,,,,, <’ffff,ffff,,,,,

**DDERR\_GENERIC**

DDERR\_HEIGHTALIGN

**DDERR\_HWNDALREADYSET****DDERR\_HWNDSUBCLASSSED****DDERR\_IMPLICITLYCREATED****DDERR\_INCOMPATIBLEPRIMARY**

## DDERR\_INVALIDCAPS

## DDERR\_INVALIDCLIPLIST

## DDERR\_INVALIDDIRECTDRAWGUID

**DDERR\_INVALIDMODE****DDERR\_INVALIDOBJECT**

## DDERR\_INVALIDPARAMS

## DDERR\_INVALIDPIXELFORMAT

## DDERR\_INVALIDPOSITION

**DDERR\_INVALIDRECT****DDERR\_INVALIDSURFACETYPE**

## DDERR\_LOCKEDSURFACES

**DDERR\_NO3D****DDERR\_NOALPHAHW**

DDERR NOBLTHW

```

 ffff“f ffff,‘ ,,,
DDERR_NOCLIPLIST
 fffffff fff,—,,,,
DDERR_NOCLIPPERATTACHED
 ,,f fff fffffff,,DirectDrawClipper fffffff,ffff,,
DDERR_NOCOLORCONVHW
 •Šf ffff,‘ ,,, ,,,,—,,,,, —,Ž ,,,
DDERR_NOCOLORKEY
 f fff,Œ ff f ,Ž,,,,
DDERR_NOCOLORKEYHW
 “ ff f ,‘,f fffffff f,,,, —,Ž ,,,
DDERR_NOCOOPERATIVELEVELSET
 IDirectDraw2::SetCooperativeLevelffff,Œ, ,,, Š ,Œ, ,,,
DDERR_NODC
 ,, ,,,,,f fff—, ,,,,,
DDERR_NODDROPSHW
 DirectDraw fff — ROP f ffff,—,,,,
DDERR_NODIRECTDRAWHW
 f ffff — DirectDraw fffffff ,,, ffff,f ffff,ff f,,
DDERR_NODIRECTDRAWSUPPORT
 Œ ,ffffff ffff,DirectDraw,ff f,.,,
DDERR_NOEMULATION
 fffffff ffff fff,—,,,,
DDERR_NOEXCLUSIVEMODE
 ffff fff,‘,”“f f,—, ,,, ffff fff,”“f f,Ž,.,,
DDERR_NOFLIPHW
 %Žf fff,ffffff,.,,
DDERR_NOGDI
 GDI,‘ ,,,
DDERR_NOHWND
 fffffff’,ffffff ffff,•—
 ,, ,,,ffffff ffff,‘fff fffff ffff,,,,,,fff,.,,
DDERR_NOMIPMAPHW
 fffffff fffff fffff,f ffff,‘ ,,,,,,—, ,,, —,Ž ,,,
DDERR_NOMIRRORHW
 ff fff f ffff,‘ ,,,,,,—, ,,, —,Ž ,,,
DDERR_NOOVERLAYDEST
 “ Š—
 ,,,IDirectDrawSurface2::UpdateOverlayffffŒ, ,,,,,f f ff,‘, IDi
 rectDrawSurface2::GetOverlayPositionffff,Œ, ,,,
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,Œ, ,,  
**DDERR\_NOTTEXTUREHW**  
ffffff fffff f ffff,‘ ,,,,,,,,,,——,,,,,, —,Ž ,,,  
**DDERR\_NOTFLIPPABLE**  
ffffff,,f fff,ffffff,,,,,,,,  
**DDERR\_NOTFOUND**  
—<,,, —,Œ,,,,,,,,  
**DDERR\_NOTINITIALIZED**  
ffffff, Š%,,,‘, CoCreateInstance, ,,, DirectDraw  
ffffff,fff ffff ffff,Œ, ,,,,  
**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  
 fffffff,,IDirectDrawSurface2::Restoreffff,E, ,,,,,,  
**DDERR\_SURFACENOTATTACHED**  
 —<,,f fff,ffff,,  
**DDERR\_TOOBIGHEIGHT**  
 DirectDraw ,,,—<,, ,‘,,,  
**DDERR\_TOOBIGSIZE**  
 DirectDraw ,,,—<,,‘,‘, ,,, E\*, ,,\*—‘,,,  
**DDERR\_TOOBIGWIDTH**  
 DirectDraw ,,,—<,,\*,‘,,,  
**DDERR\_UNSUPPORTED**

---

*ff f,,,,, —,,,*

**DDERR\_UNSUPPORTEDFORMAT**

*—<sub>4</sub>,,, FourCC *ff fff*,DirectDraw <sub>4</sub>,,,*ff f*,,,,,*

**DDERR\_UNSUPPORTEDMASK**

*—<sub>4</sub>,,,*ffff ff fff*,,*ffffff* ,DirectDraw <sub>4</sub>,,,*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**

*—,,,⌘,•—,,,”^, •^—,•,,,,,,,,,*