Microsoft® DirectX™ 3
ƒƒƒƒƒƒŠ”ƒƒƒ

,,ffffff,‹ ,,,,, •, — ,,,• ,,,,,,,, ,, •’,‰Ž– –
,,,f f,“,‹ ,,,,,,,,,‰‹,,,,,, ,,ffffff,‘•,,,ˆ•, Microsoft Corporation, –,,,‹‰
,,,, “‹“ ‹Š“,,,,,,•Ž •–,,,• “’,,,,,,,,,
Microsoft , ,,ffffff,‹ ,,,,, ffff fff,“‹,,,, ’,“‹ • ’ Œ ,,‘,’“ —
Œ,•Ž,,,, ,,ffffff, Microsoft , –,,,‹‰,,,, “‹ • ’ Œ ,,‘,’“ —Œ,‹‰,—
,,,,,,,,

©1996 Microsoft Corporation. All rights reserved.

Microsoft ActiveMovie Direct3D DirectDraw DirectInput DirectPlay DirectSound DirectX MS-DOS Win32 Windows ,,,Windows NT, • Microsoft Corporation,• ,,,,,‘, ,,,, •,,,

,,‘, •–,‰Ž–,ŠŽ, •,,,

2

DirectDraw

- DirectDraw„„
- DirectDrawƒ□ƒƒƒƒƒ
- DirectDrawŠ—
  - DirectDraw
  - „‘,DirectDraw,“’
  - DirectDraw HAL
  - DirectDraw HEL
  - DirectDrawƒƒƒƒƒƒ,ƒƒƒ
  - •,ƒƒƒ
  - 3Dƒ□ƒƒƒ,ƒƒ□ƒ
  - DirectDraw,Direct3D,“□
  - ƒ□ƒXƒƒƒƒƒƒ□ƒ□ƒ
  - ƒƒƒƒ□ƒƒ□ƒƒƒ
- DirectDrawƒƒƒ ƒƒƒƒŠ—
  - IDirectDraw2ƒƒƒ ƒƒƒƒ
  - IDirectDrawClipperƒƒƒ ƒƒƒƒ
  - IDirectDrawPalette ƒƒƒ ƒƒƒƒ
  - IDirectDrawSurface2ƒƒƒ ƒƒƒƒ
- DirectDraw ƒƒ ƒƒƒƒ
  - Tutorial 1: DirectDraw,Š‘
  - Tutorial 2: ƒƒƒƒƒƒ,ƒƒƒ□ƒƒƒƒ,“‘
  - Tutorial 3: ƒƒƒƒƒ□ƒ□ƒ□ƒƒƒ„„ƒƒƒƒ“‘
  - Tutorial 4: ƒƒ□¥ƒ□,ƒƒƒƒƒƒƒƒƒ□ƒƒƒ
  - Tutorial 5: ƒƒƒƒ,““•□
  - „‘, DirectDrawƒƒƒƒ
  - □“‰,ƒƒƒƒƒƒ
- DirectDrawƒƒƒƒƒ
  - Š□
  - ƒ ƒƒƒƒŠ□
  - 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

- ƒƒƒƒƒƒƒ,ƒƒƒƒƒƒ ƒƒƒƒƒƒƒ,ƒƒ ƒ
- ƒƒƒƒƒƒ¥ƒ ƒ,ƒ ƒƒƒƒ ƒƒƒƒ“‘,,ƒƒƒƒ, Œ
- 3D Zƒƒƒƒ ,ƒƒ ƒ
- Zƒ ƒƒƒƒ,ƒ ƒƒƒƒ ƒ ƒ ƒƒ,ƒƒ ƒ
- ƒƒ ƒ ’ƒ ƒƒƒƒ,,ƒƒƒƒ
- ƒƒƒƒ ƒ,,,ƒƒƒƒƒƒ‘•,ƒƒƒƒƒƒ¥ƒƒƒ—ˆ,,“Žƒƒƒƒ

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

Windows 95 Windows NT® version 4.0
DirectDraw の

DirectDrawƒ□ƒƒƒƒƒ

DirectDraw ¥ ¥

DirectDraw
DirectDraw の (HAL)
¥

DirectDrawŠ—

のDirectDraw の

- *DirectDraw*
- *,,‘, DirectDraw,“’*
- *DirectDraw HAL*
- *DirectDraw HEL*
- *DirectDrawƒƒƒƒƒƒ,ƒƒƒ*
- *•,ƒƒƒ*
- *3D ƒ ƒƒƒ,ƒƒ ƒ*
- *DirectDraw,Direct3D,“*
- *ƒ ƒXƒƒƒƒƒƒ ƒ ƒ*
- *ƒƒƒƒ ƒƒ ƒƒƒ*

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¥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 DirectDraw
¥ ¥のの Windows 95 Windows NT
GDI **CreateDIBSection**
DirectDraw

の DirectDraw DirectDrawPalette ¥
¥ の
IDirectDrawPalette
Windows ¥
DirectDrawPalette DirectDraw

DirectDraw DirectDrawClipper DirectDraw
¥の

,,‘,DirectDraw,“’

DirectDraw DirectDraw

¥
¥
¥

¥

DirectDraw ¥
のの
のの

DirectDrawSurface ¥
¥ ¥

¥
Z

DirectDraw HAL

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

DirectDraw HAL HAL
の
HAL HAL
DirectDraw

DirectDraw HEL

DirectDraw ¥ ¥ (HEL) HAL
DirectDraw DirectDraw HAL
DirectDraw

DirectDrawƒƒƒƒƒƒ‚ƒƒƒ

DirectDraw
DirectDraw DirectDraw
DirectDraw

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

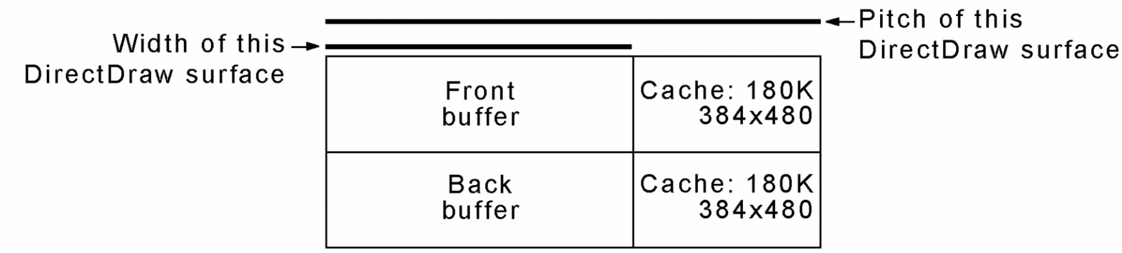
DirectDrawPalette 16 256 の¥

DirectDraw DirectDrawSurface DirectDrawPalette
DirectDrawClipper の DirectDrawPalette DirectDrawClipper
DirectDrawSurface
DirectDrawSurface DirectDrawPalette
の

•‚ƒƒƒ

¥
幅とピッチ のののの
のののののののののののののの

¥のののの


Width of this DirectDraw surface
Pitch of this DirectDraw surface
Front buffer
Cache: 180K 384x480
Back buffer
Cache: 180K 384x480

の¥ 640×480×8 384×480×8 の
640 384 1024

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

¥

3Dƒ□ƒƒƒ‚ƒƒ□ƒ

の DirectDraw 3D

- *ƒƒƒƒƒ ƒƒƒ*
- *ƒƒƒƒƒƒ*
- *Zƒƒƒƒ*

ƒƒƒƒƒ¥ƒƒƒ

テクスチャ¥マップ HEL テクスチャ¥マップ¥ **IDirectDraw2::CreateSurface** **DDSURFACEDESC** の **ddsCaps** **DDSCAPS_TEXTURE**

HEL ¥ ¥のの

ƒƒƒƒƒƒ

DirectDraw ¥ ¥
ののの

の

の

DirectDraw
ののの

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;
LPDIRECTDRAWSURFACE2 lpDDMipMap;
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

```
LPDIRECTDRAWSURFACE lpDDLevel, lpDDNextLevel;
DDSCAPS ddsCaps;

lpDDLevel = lpDDMipMap;
lpDDLevel->AddRef();
ddsCaps.dwCaps = DDSCAPS_TEXTURE | DDSCAPS_MIPMAP;
ddres = DD_OK;
while (ddres == DD_OK)
{
    // ,,ƒƒƒ,□–
    .
    .
    .
    ddres = lpDDLevel->GetAttachedSurface(
        &ddsCaps, &lpDDNextLevel);
    lpDDLevel->Release();
    lpDDLevel = lpDDNextLevel;
}
if ((ddres != DD_OK) && (ddres != DDERR_NOTFOUND))
.
.
.
```

¥の¥ ¥
¥の
の **DDSCAPS_MIPMAP**
DDSCAPS_TEXTURE
)

DDSCAPS_MIPMAP ¥
¥ **DDSCAPS_BACKBUFFER** の

のの **IDirectDraw2::CreateSurface** ¥

IDirectDrawSurface2::AddAttachedSurface ¥
¥
IDirectDrawSurface2::AddAttachedSurface

Note *ffff“‘‘ , ffffff ff f,ˆfff,,,“—*
,,, ffffff,‘ff f,ffff“‘,,,,,, Šfff,• ,ffff“‘,,,,,,,,,

IDirectDrawSurface2::Flip のの
フリッピング フリッピング ののの
¥ ¥ フリッピング ¥
のの¥ののの¥ フリッピング

IDirectDrawSurface2::Lock **IDirectDrawSurface2::GetSurfaceDesc**
)
DDSURFACEDESC
dwMipMapCount
dwMipMapCount

Zffff

DirectDraw HEL Direct3D™ 3D ¥ Z
HEL 16 32 の Z
3D の DirectDraw ¥
DDSCAPS_ZBUFFER ¥
Z の ¥ **DDCAPS** の

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

DirectDraw ¥ Z の
DDCAPS_BLTDEPTHFILL **DDBLT_DEPTHFILL**
の Z

Note ‰ ,’,Ž ,‰Ž, 3Dƒƒƒƒƒƒ ƒƒƒƒ,ˆ‘,,

DirectDraw,Direct3D,“□

のDirectDraw Direct3D

- *Direct3Dƒƒƒƒ ƒƒƒ ƒƒƒƒ*
- *Direct3D ƒƒƒƒ ƒƒƒ ƒƒƒƒ*
- *Direct3D ƒƒƒƒƒ ƒƒƒ ƒƒƒƒ*
- *DirectDraw HEL, Direct3D*

Direct3D*ƒƒƒƒ*□*ƒƒƒ*□*ƒƒƒƒ*

DirectDraw
DirectDraw Direct3D の DirectDraw の COM
Direct3D の COM
Direct3D Direct3D
DirectDraw Direct3D
COM **QueryInterface**

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 ドライバ・ス
テータス DirectDraw
Direct3D Direct3D ドライバ¥
DirectDraw
Direct3D DirectDraw ドライバ Direct3D
Direct3D

Direct3Dƒƒƒƒ□ƒƒƒ□ƒƒƒƒ

Direct3D Direct3D
Direct3D 3D DirectDrawSurface

DirectDrawSurface Direct3D

```
LPDIRECTDRAWSURFACE lpDDSurface;
LPDIRECT3DDEVICE    lpD3DDevice;

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

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

Direct3Dƒƒƒƒƒ□ƒƒƒ□ƒƒƒƒ

Direct3D DirectDrawSurface の
DirectDrawSurface Direct3D

```
LPDIRECTDRAWSURFACE lpDDSurface;
LPDIRECT3DTEXTURE   lpD3DTexture;

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

.
.

Direct3D
(*Direct3D*)
DirectDrawSurface

DirectDraw HEL,Direct3D

DirectDraw HEL Zの
DirectDraw Direct3Dの DirectDraw
Direct3D
DirectDraw HEL 3D
DDSCAPS_3DDEVICE
3D ¥の DirectDraw
3D

ƒ□ƒXƒƒƒƒƒƒ□ƒ□ƒ

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

Xの
.

X Windows NT

ƒƒƒƒ□ƒƒ□ƒƒƒ

の ¥ ¥ (HEL)
¥ の

- *ƒƒƒƒƒ ƒƒƒ,ƒƒ ƒƒƒ*
- *ƒƒƒƒƒ ƒ ƒ ƒƒƒ,ƒƒ ƒƒƒ*

ƒƒƒƒƒ□ƒƒƒ,ƒƒ□ƒƒƒ

HELののred green blue alpha
のの

ƒƒƒƒ ƒƒ ƒƒƒ ƒƒƒ	ƒƒƒ “	ƒƒƒ
DDPF_RGB |	1	R: 0x00000000
DDPF_PALETTEINDEXED1		G: 0x00000000
		B: 0x00000000

		A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED1 | DDPF_PALETTEINDEXEDTO8	1	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED2	2	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED2 | DDPF_PALETTEINDEXEDTO8	2	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED4	4	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED4 | DDPF_PALETTEINDEXEDTO8	4	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED8	8	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB	8	R: 0x000000E0 G: 0x0000001C B: 0x00000003 A: 0x00000000
DDPF_RGB | DDPF_ALPHAPIXELS	16	R: 0x00000F00 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 | DDPF_ALPHAPIXELS	32	R: 0x00FF0000 G: 0x0000FF00 B: 0x000000FF A: 0xFF000000
DDPF_RGB | DDPF_ALPHAPIXELS	32	R: 0x000000FF G: 0x0000FF00 B: 0x00FF0000 A: 0xFF000000

HEL 3D のの DirectDraw

DDSCAPS_TEXTURE

ƒƒƒƒƒ□ƒ□ƒ□ƒƒƒ‚ƒƒ□ƒƒƒ

DirectX 3 HEL ののred green blue alpha のの

ƒƒƒƒ ƒƒ ƒƒƒ ƒƒƒ	ƒƒƒ “	ƒƒƒ
DDPF_RGB | DDPF_PALETTEINDEXED1	1	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED2	2	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED4	4	R: 0x00000000 G: 0x00000000 B: 0x00000000 A: 0x00000000
DDPF_RGB | DDPF_PALETTEINDEXED8	8	R: 0x00000000 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

DirectDrawƒƒƒ□ƒƒƒƒŠ—

の DirectDraw COM

- *IDirectDraw2* ƒƒƒ ƒƒƒƒ
- *IDirectDrawClipper* ƒƒƒ ƒƒƒƒ
- *IDirectDrawPalette* ƒƒƒ ƒƒƒƒ
- *IDirectDrawSurface2* ƒƒƒ ƒƒƒƒ

IDirectDraw2ƒƒƒ□ƒƒƒƒ

の IDirectDraw2

- *DirectDrawƒƒƒƒƒƒ*
- *IDirectDraw2, ‰, ,,,,,,?*
- *—,,,• DirectDrawƒƒƒƒƒƒ*
- *‰‘“ ƒƒƒ ƒƒ ƒƒƒ “,ƒƒ ƒ*
- *ƒƒƒƒƒ ƒ ƒƒƒ,ƒƒ ƒ‹—ƒƒƒ*
- *ƒ ƒ• ,”‘“ƒƒƒƒ*
- *CoCreateInstance,—,,DirectDrawƒƒƒƒƒƒ,*

DirectDrawƒƒƒƒƒƒ

DirectDraw オブジェクト

DirectDraw
DirectDrawSurface DirectDrawPalette DirectDrawClipper

DirectDraw オブジェクト
Windows 95 Windows 95

DirectDraw HAL Windows 95 GDI の DirectDraw オブジェクト
(GUID) DirectDraw オブジェクト GUID
DirectDrawEnumerate

DirectDraw オブジェクト
8 / ¥
¥ ¥の

IDirectDraw2,‰,□,,,,,,?

DirectX COM
IDirectDraw2 **IDirectDraw**
IDirectDraw2 **IDirectDraw::QueryInterface**

```
// IDirectDraw2ƒƒƒ□ƒƒƒƒ,□□
LPDIRECTDRAW  lpDD;
LPDIRECTDRAW2 lpDD2;

ddrval = DirectDrawCreate(NULL, &lpDD, NULL);
if(ddrval != DD_OK)
    return;

ddrval = lpDD->SetCooperativeLevel(hwnd,
    DDSCL_NORMAL);
```

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

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

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

C++の**IDirectDraw2**
IDirectDraw::QueryInterface
IDirectDraw2::GetAvailableVidMem **IDirectDraw**

IDirectDraw2::GetAvailableVidMem **IDirectDraw2**
IDirectDrawtDraw
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
のXの

IDirectDraw
IDirectDraw2

□—,,,•□DirectDrawƒƒƒƒƒ

DirectDraw **DirectDrawCreate**

DirectDraw オブジェクト

DirectDraw オブジェクト DirectDraw オブジェクト
DirectDrawSurface DirectDrawPalette DirectDrawClipper のオブジェクト DirectDraw オブジェクト

DirectDraw □□□□□□
DirectDraw □□□□□□

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

□‰‘“‚ƒƒƒ□ƒƒ□‚ƒƒƒ□“ƒƒ□ƒ

DirectDraw の
DirectDraw のの 24-bpp 32-bpp

DirectDraw HEL 24- bpp 32-bpp の
の
¥
HEL

Windows DirectDraw

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

ƒƒƒƒƒ□ƒ□ƒƒƒ¥ƒƒ□ƒ‹—ƒƒƒ

DirectDraw

の

DirectDraw GDI の
GDI GDI の
DirectDraw GDI
GDI の DirectDrawSurface □□□□□□
IDirectDraw2::GetGDISurface

GDI DirectDraw ¥
(DDB) DirectDraw の
DirectDraw の

HAL DIB ¥

ƒ□ƒ•□‚”‘“ƒƒƒƒ

IDirectDraw2::SetDisplayMode
¥

の DirectDraw □□□□□□

DirectDrawSurface オブジェクト ¥
IDirectDrawSurface2::Restore

DirectDraw
DirectDrawSurface オブジェクト DirectDraw GDI のの

CoCreateInstance,—,,Directdrawƒƒƒƒƒƒ,□□

DirectDrawCreate **CoCreateInstance**
IDirectDraw2::Initialize DirectDraw

1 ƒƒƒƒ ƒƒƒ,Ž,, NULL,Ž',,**CoInitialize**,Œ, , COM, Š‰,,

```
if (FAILED(CoInitialize(NULL)))
    return FALSE;
```

2 **CoCreateInstance**,**IDirectDraw2::Initialize**
ƒƒƒƒ,Ž,,DirectDrawƒƒƒƒƒƒ, ,,

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

,,**CoCreateInstance**,Œ, ,' ,ƒƒƒ ƒ*CLSID_DirectDraw*, DirectDrawƒƒƒƒ ƒƒƒƒƒƒ ƒƒƒ,ƒƒƒŽ•Ž,,, *IID_IDirectDraw2* ƒƒƒ ƒ, ,,,"',DirectDraw ƒƒƒ ƒƒƒƒ,Ž•,, *lpdd*ƒƒƒ ƒ, Ž",,,DirectDrawƒƒƒƒƒƒ,Ž, Œ, ,, Œ,,, ,,Š , Š‰ ,,,,,,ƒƒƒƒƒƒ,•,

3 DirectDrawƒƒƒƒƒƒ,Ž,', **IDirectDraw2::Initialize**,Œ, ,,,,,,,,, ,,ƒƒƒƒ, **DirectDrawCreate** Š ,ˆ"",Ž,ƒƒƒƒ GUID ƒƒƒ ƒ ,, NULL ,ˆ ,Ž, DirectDrawƒƒƒƒƒƒ Š‰Œ, **DirectDrawCreate** Š ,Ž,, ,,,,"—,Ž— ‰•,,,,,,,, DirectDrawƒƒƒƒƒƒ,Š˜,,,,,ƒƒƒƒ,Ž,', **IDirectDraw2::Initialize** ƒƒƒƒ,Œ, ,,,, DDERR_NOTINITIALIZEDƒƒ ," ,, .

CoUninitialize COM

```
CoUnitialize();
```

IDirectDrawClipperƒƒƒ□ƒƒƒƒ

IDirectDrawClipper

IDirectDrawClipper

- *ƒƒƒƒƒƒ ƒƒƒ*
- *DirectDrawClipperƒƒƒƒƒƒ‚‹—*
- *ƒƒƒƒ”ˆ‘DirectDrawClipperƒƒƒƒƒƒ*
- *CoCreateInstance‚—‚‚DirectDrawClipperƒƒƒƒƒƒ‚*

ƒƒƒƒƒƒ□ƒƒƒ

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

DirectDrawClipper*ƒƒƒƒƒƒ‚‹—*

DirectDrawClipper オブジェクト
DirectDrawClipper オブジェクト
¥の **DirectDrawSurface2::SetClipper** DirectDrawClipper オブジェクトを
0 DirectDrawClipper オブジェクト ¥
¥ NULL **IDirectDrawSurface2::SetClipper**
DirectDrawClipper オブジェクト
DirectDrawClipper オブジェクト

Note “‚DirectDrawClipper*ƒƒƒƒƒƒ* “‚*ƒ ƒƒƒ*‚‘‚ **IDirectDrawSurface2::SetClipper**‚‰“‚‘‚‚Œ‚ ‚‚ *ƒƒƒƒƒƒ*‚Ž *ƒƒƒƒ*‚ˆ “‚‚*ƒƒƒƒƒƒƒ*‚‚‚ ˜‘‚‚Œ‚ ‚‚ *ƒƒƒƒƒƒ*‚Ž *ƒƒƒƒ*‚‰‹‚—‚‚‚‚‚‚

ƒƒƒƒ”ˆ‘‚DirectDrawClipper*ƒƒƒƒƒƒ*

DirectDraw DirectDrawClipper
DirectDrawClipper DirectDraw
DirectDrawClipper **DirectDrawCreateClipper**
DirectDraw
DirectDawClipper

DirectDraw DirectDrawClipper
DirectDraw

IDirectDraw2::CreateClipper DirectDrawClipper
DirectDrawClipper DirectDraw

CoCreateInstance,— ,,DirectDrawClipperƒƒƒƒƒ,□□

DirectDrawClipper オブジェクト COM のの
DirectDrawCreateClipper **IDirectDraw2::CreateClipper**
CoGetClassObject
CoCreateInstance **CoCreateInstance**
DirectDrawClipper オブジェクト
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 オブジェクト・クラス *IID_IDirectDrawClipper*
lpClipper
DirectDrawClipper オブジェクト

の DirectDrawClipper オブジェクトを使用前に
IDirectDrawClipper::Initialize 0UL *dwFlag*
0
lpDD DirectDrawClipper オブジェクト DirectDraw
NULL の DirectDrawClipper オブジェクト
DirectDrawCreateClipper DirectDrawClipper

CoUninitialize COM

```
CoUnitialize();
```

IDirectDrawPalette ƒƒƒ□ƒƒƒƒ

IDirectDrawPalette

- *DirectDrawPaletteƒƒƒƒƒƒ*
- *”ƒƒƒƒƒ ƒ ƒƒƒ ,,ƒƒƒƒ ’*
- *ƒƒƒƒ,‹—*
- *ƒƒƒƒ¥ƒƒƒ*

DirectDrawPaletteƒƒƒƒƒƒ

DirectDrawPalette 16 256
DirectDrawPalette DirectDrawSurface

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

IDirectDrawPalette::GetEntries
IDirectDrawPalette::SetEntries
の *dwFlags*

DirectDrawPalette
のの の

IDirectDrawPalette::SetEntries
DirectDrawPalette
DirectDrawSurface DirectDrawPalette

”ƒƒƒƒƒ□ƒ□ƒƒƒ„ƒƒƒƒ‚□’

¥の
DirectDraw

IDirectDrawSurface2::GetDC
DirectDrawPalette

Direct3D (3D)
¥

ƒƒƒƒ‚‹—

¥ ¥
IDirectDrawSurface2::SetPalette

0
¥ ¥ NULL

Note “‚ƒƒƒƒ “‚ƒ ƒƒƒ‚‘‚
IDirectDrawSurface2::SetPalette‚‰“‚‘‚‚Œ‚ ‚‚‚ ƒƒƒƒ‚Ž ƒƒƒƒ‚ˆ“‚
‚ƒƒƒƒƒƒƒ‚‚‚‚ ˜‘‚‚Œ‚ ‚‚ ƒƒƒƒ‚Ž ƒƒƒƒ‚‰‹‚—‚‚‚‚‚‚‚

ƒƒƒƒ□ƒƒƒ

DirectDraw 1 (2) 2 (4) 4 (16)
8 (256) ¥
DDPCAPS_1BIT

2 **DDPF_PALETTEINDEXED1**
1 .

RGB **PALETTEENTRY**
2 4

IDirectDraw2::CreatePalette .
DDPCAPS_8BITENTRIES の
DDPCAPS_4BIT |
DDPCAPS_8BITENTRIES
PALETTEENTRY の
IDirectDraw2::CreatePalette
LPPALETTEENTRY

ƒƒƒƒƒ□*ƒ*□*ƒ*,,DirectDraw*ƒƒƒƒ*Ž—

IDirectDrawPalette

IDirectDrawPalette GDI
DirectDraw
の

- (**IDirectDraw2::CreatePalette**,**IDirectDrawPalette::SetEntries***ƒƒƒƒ*,“,,,) **PALETTEENTRY** ‘‘,**peFlags***ƒƒƒ*,“Š,*ƒƒƒ*,,,,,,,,,
- Windows, “*ƒƒƒƒ ƒƒƒƒ* 0,,9 246,,255,*ƒƒƒƒƒƒ*,,,,,),• ,,,,,,,,,,

の Windows のWindows 8

- *ƒƒƒƒƒ ƒ ƒ*,*ƒƒƒƒ ƒƒƒƒ ƒƒƒ*
- *ƒƒƒƒƒ ƒ ƒ*,,IDirectDraw::CreatePalette,Œ, ,
- *ƒƒƒƒƒ ƒ ƒ*,IDirectDrawPalette::SetEntries,Œ, ,

ƒƒƒƒƒ□*ƒ*□*ƒ*,*ƒƒƒƒ*□*ƒƒƒƒ*□*ƒƒƒ*

PALETTEENTRY の **peFlags**
IDirectDraw2::CreatePalette **IDirectDrawPalette::SetEntries**
peFlags

3

- *Windows* “*ƒƒƒƒ*

  ˆ”“, Windows, 0,,9 246,,255,,,*ƒƒƒƒ*,——
  ƒƒƒƒƒƒƒƒ ,,,, *ƒƒƒƒ ƒƒƒ*,,,,*ƒƒƒƒ*, ’,• ,,,,,‹Ž,,,, ,,,, ,Š,256 *ƒƒƒƒ ƒƒƒƒ*,**GetSystemPaletteEntries** Win32
  Š ,Œ, ,, ,,,,,,,,, *ƒƒƒƒ ƒƒƒ*,’ ,,,,*ƒƒƒƒ*,Ž ,,•,Œ—“,,,
  ,,,,,, *ƒƒƒƒ ƒ ƒƒ*,—,,,,*ƒƒƒƒ*,,,•

—“,ƒƒƒƒ ƒƒƒƒƒƒ,ƒƒƒ,,,Ž’,, PC_EXPLICIT ƒƒƒ,**PALETTEETRY** ‘‘,**peFlags**ƒƒƒ ,Š”, ƒƒƒƒ,ƒƒƒ,,•— ƒƒƒƒ,ƒƒƒƒƒƒ,,,**peRed**ƒƒƒ,ƒƒƒ,, ,,,,ƒƒƒƒ ƒƒƒ,,’ Ž ,,,,,,,, ƒƒƒƒ ƒƒƒ,,, ‹ “,Windows,0,255,ƒƒƒƒ,,,──,,,,,,,,,, **SetSystemPaletteUse** Win32 Š ,Œ,,,,,,,, ,, ,PC_EXPLICIT,‘,,**PALETTEENTRY** ‘‘,0,255,ƒƒƒƒ,,,Ž’,,,,,,,

- *ƒƒƒ ƒƒƒ ƒƒƒƒ*

  ƒƒƒƒ ƒƒƒ, ƒƒƒƒ ƒƒƒ ƒƒƒŒ‰, ,,,,,• ,,, ƒƒƒ ƒƒƒ ƒƒƒƒ,,,, PC_RESERVED ƒƒƒ,ƒƒƒƒ ƒƒƒ,Ž’,, Windows,‘ƒƒƒƒ ƒƒƒ,˜—ƒƒƒƒ,•— ƒƒƒƒ,ƒƒƒ,,,,,‹Ž,, ,,,,, ƒƒƒƒ ƒƒƒ,ƒƒƒƒ,ˆ,, , ’,,,,, ‘,ƒƒƒƒ ƒƒƒ,,, ,– ,,,,,,,,”,,,,

- *”ƒƒƒ ƒƒƒ ƒƒƒƒ*

  ƒƒƒƒ ƒƒƒ,• ,,,,ƒƒƒƒ,,, ”ƒƒƒ ƒƒƒ ƒƒƒƒ, ’,PC_NOCOLLAPSE ƒƒƒ,– ,,,,,, ,,ƒƒƒ, ‘,Š,“, ,•—ƒƒƒƒ ƒƒƒƒ,Windows,’,Š,,,,,,,,,

PALETTEENTRY の 3

ƒƒƒƒ ƒƒƒ	**peFlags’**	**peRed, peGreen, peBlue’**
Windows “ƒƒƒƒ(0-9, 246-255 ,,,0,255,,ƒƒƒƒƒƒ)	PC_EXPLICIT	peRed = index, peGreen = 0, peBlue = 0
ƒƒƒ ƒƒƒ ƒƒƒƒ	PC_RESERVED | PC_NOCOLLAPSE	’
”ƒƒƒ ƒƒƒ ƒƒƒƒ	PC_NOCOLLAPSE	’

ƒƒƒƒƒ□ƒ□ƒ,,IDirectDraw2::CreatePalette,Œ,□,

DirectDraw **IDirectDraw2::CreatePalette** **PALETTEENTRY** の 256

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

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

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

// ƒƒƒƒ□ƒƒƒƒƒ,ƒƒƒƒƒƒ□,,–,,□——,,,
// □□,16ƒƒƒƒ,ƒƒƒ□ƒƒƒ,,,
for (index = 10; index < 26; index ++)
{
    pPaletteEntry[index].peFlags = PC_NOCOLLAPSE|PC_RESERVED;
    pPaletteEntry[index].peRed = 255;
    pPaletteEntry[index].peGreen = 64;
    pPaletteEntry[index].peBlue = 32;
}

// Ž,,ƒƒƒƒƒƒ□”ƒƒƒ□ƒƒƒ□ƒƒƒƒ
for (; index < 246; index ++) // ‘,ƒ□ƒ,ƒƒƒƒƒƒ,ƒƒƒ
{
    pPaletteEntry[index].peFlags = PC_NOCOLLAPSE;
    pPaletteEntry[index].peRed = 25;
    pPaletteEntry[index].peGreen = 6;
    pPaletteEntry[index].peBlue = 63;
}

// ,,,,256ƒƒƒƒ,–,,,□ƒƒƒƒ,□□
ddrval = lpDD->CreatePalette(DDPCAPS_8BIT, pPaletteEntry,
    &lpDDPal,NULL);
```

ƒƒƒƒƒ□ƒ□ƒ,,IDirectDrawPalette::SetEntries,Œ,□,

IDirectDrawPalette::SetEntries **PALETTEENTRY** **IDirectDraw2::CreatePalette** **PALETTEENTRY** の **IDirectDrawPalette::SetEntries**

の Windows の Windows の の 256 **PALETTEENTRY** の **IDirectDrawPalette::SetEntries** PC_NOCOLLAPSE PC_RESERVED

のの

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

// ,,,,,ƒƒƒƒ,ƒƒƒ□ƒƒƒ,,□,,,,16,——,,,
// ƒƒƒƒ,□Š,,,□,,,,,,ƒƒƒ□ƒƒƒ,□,,,
temp = pPaletteEntry[10];
for (index = 10; index < 25; index ++)
```

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

// ',□',,□ƒƒƒƒ□ƒƒƒƒ□'''',,ƒƒƒƒ,,,,□
// •□,,,ƒƒƒƒ,,,,ƒƒƒƒ,‟,,,,,,,,,
ddrval = lpDDPal->SetEntries(
    0,                      // ƒƒƒ,ƒƒ,,,,,,,,,
    10,                     // □□,ƒƒƒƒ
    16,                     // ƒƒƒƒ□
    & (pPaletteEntry[10])); // ƒ□ƒ,‟,,,,,,
```

IDirectDrawSurface2*ƒƒƒ□ƒƒƒƒ*

IDirectDrawSurface2

- *DirectDrawSurfaceƒƒƒƒƒƒ*
- *IDirectDrawSurface2,‰, ,,,,,,?*
- *ƒ ƒƒƒ,*
- *ƒƒ ƒ¥ƒƒƒƒ,ƒƒƒƒ*
- *ƒƒƒƒƒƒ ƒ ƒƒƒ ,GDI,ƒƒ ƒ¥ƒ ƒ*
- *ƒ ƒƒƒ, Ž*
- *•Š,ƒƒ ƒƒƒ•Š*
- *ƒƒ ¥ƒ*
- *Zƒ ƒ ƒ ƒ ƒƒ*
- *ƒƒƒƒƒ ƒ ƒ ƒƒƒ,,,,ƒƒƒƒƒ ƒƒƒƒ*
- *ƒƒƒƒ ¥ƒƒƒ¥ ƒ ƒƒƒ,, ƒƒƒƒ“‘*

DirectDrawSurface*ƒƒƒƒƒƒ*

DirectDrawSurface 2D の¥ DirectDrawSurface DirectDraw DirectDrawSurface **IDirectDraw2::CreateSurface** DirectDrawSurface ¥の¥ DirectDraw DirectDrawSurface の DirectDrawSurface

DirectDrawSurface ¥のの CPU

DirectDrawSurface Windows ¥ DirectDrawSurface GDI Windows GDI ¥ の (HDC) DirectDrawSurface GDI HDC ¥ HDC

IDirectDrawSurface2,‰,□,,,,,,?

DirectX COM

IDirectDrawSurface2 **IDirectDrawSurface** **IDirectDraw::QueryInterface**

```
LPDIRECTDRAWSURFACE  lpSurf;
LPDIRECTDRAWSURFACE2 lpSurf2;

// ƒ□ƒƒƒ,□□.
memset(&ddsd, 0, sizeof(ddsd));
ddsd.dwSize = sizeof(ddsd);
ddsd.dwFlags = DDSD_CAPS | DDSD_WIDTH | DDSD_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 DirectDrawSurface
DDSCAPS 構造体の**dwCaps** DDSCAPS_FLIP

IDirectDraw2::CreateSurface
DirectDrawSurface

IDirectDrawSurface2::DeleteAttachedSurface

¥ ¥

の

ƒƒƒƒ 1

GDI DirectDrawSurface
DDSURFACEDESC 構造体の **dwHeight** **dwWidth** のののの
の

DDSURFACEDESC 構造体 (
ddsd)

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

// ,,ƒƒƒ,—Œ,DirectDraw,’,,,
ddsd.dwFlags = DDSD_CAPS;

// ƒƒƒƒƒ□ƒ□ƒƒƒ,ƒƒƒƒƒ
ddsd.ddsCaps.dwCaps = DDSCAPS_PRIMARYSURFACE;
```

ƒƒƒƒ 2

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

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

// ,,ƒƒƒ,—Œ,DirectDraw,’,,,
ddsd.dwFlags = DDSD_CAPS | DDSD_HEIGHT | DDSD_WIDTH;

// Š’,ƒƒƒƒƒ□ƒ□ƒ□ƒƒƒ,—‹,,□
// ƒƒƒ,100□100ƒƒƒƒ,,,□
ddsd.ddsCaps.dwCaps = DDSCAPS_OFFSCREENPLAIN;
dwHeight = 100;
dwWidth = 100;
```

DirectDraw の
DDSCAPS 構造体の dwCaps メンバ
DDSCAPS_SYSTEMMEMORY

¥ DDSCAPS_VIDEOMEMORY

DirectDraw
IDirectDraw2::CreateSurface
IDirectDraw2::CreateSurface DDSCAPS_COMPLEX
のの
DirectDraw の
IDirectDraw::Release
IDirectDrawSurface2::Restore

ƒƒƒƒ 3

のの¥ の **DDSURFACEDESC** 構造体 (**ddsd**) ¥

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

// ,,ƒƒƒ,–Œ,DirectDraw,’,,,
ddsd.dwFlags = DDSD_CAPS | DDSD_BACKBUFFERCOUNT;

// ,,,,ƒƒƒ¥ƒƒƒƒ,,,,ƒƒƒƒƒ□ƒ□ƒƒƒ,–‹,,
//
ddsd.ddsCaps.dwCaps = DDSCAPS_COMPLEX | DDSCAPS_FLIP |
DDSCAPS_PRIMARYSURFACE;
ddsd.dwBackBufferCount = 1;
```

IDirectDrawSurface2::Flip
¥ の **DDSURFACEDESC** 構造体の
BackBufferCount メンバ 2 2 の
IDirectDrawSurface2::Flip
3 のの

ƒƒ□ƒ¥ƒƒƒƒ,ƒƒƒƒ

DirectDraw DirectDrawSurface
IDirectDrawSurface2::Lock **RECT** 構造体
IDirectDrawSurface2::Lock **RECT** 構造体 NULL
IDirectDrawSurface2::Lock

DDSURFACEDESC 構造体 の

DirectDrawSurface
の

- ˆ’,ƒƒƒƒƒƒ ƒƒƒ,‘’,,,,,,,, ,,,**IDirectDrawSurface2::Lock** ƒƒƒƒ,•,ƒƒƒ •,’,,,, ƒƒƒ, ƒ ƒƒƒ ƒƒƒ, ƒƒƒƒƒƒ¥ƒ ƒ,ƒƒƒ DirectDrawƒƒƒƒ,ƒ ƒƒƒ,,,,,,,,,——,,•‰,,

- **IDirectDrawSurface2::Lock**,Œ, ,,**IDirectDrawSurface2::Unlock**
  ƒƒƒƒ,Š,,,Ž,, Œ,,,, ƒ ƒƒƒ ƒƒƒ,ƒƒƒƒŽ“,ˆ‘, ,,,
  IDirectDrawSurface2::Lock
  ƒƒƒƒ,WIN16,ƒƒƒ,•Ž,,,, **IDirectDrawSurface2::GetDC**
  ƒƒƒƒ,ˆ–“,**IDirectDrawSurface2::Lock**,Œ, , WIN16ƒƒƒ,
  IDirectDrawSurface2::Lock, **IDirectDrawSurface2::Unlock**,Š, GDI,USE
  R,ƒƒƒƒ,**IDirectDrawSurface2::GetDC**, **IDirectDrawSurface2::ReleaseDC**
  ,Š,“,,,, Œ,,
- ƒƒƒƒƒƒ¥ƒƒƒ,˜‘,,,ƒƒ ,,,,
  ƒƒƒ¥ƒƒƒƒ¥ƒƒƒ,Ž,,ƒƒƒƒƒƒ¥ƒ ƒ,,,,‰‘ƒƒƒƒ¥ƒƒ ƒ¥ƒƒƒƒ,Ž‘,,,,, Windows
  95,ƒ ƒ ƒƒ ƒƒ ƒƒƒƒ Vflatd.386,Ž,
  ,,ƒƒƒƒ,,,, ,,,,ƒƒƒƒƒƒ ƒƒƒƒ, DirectDraw,˜‘,,ƒƒ ƒƒƒƒƒ,“,,,,,,, •’,,ƒƒƒƒƒƒ
  ¥ƒƒƒ,ƒƒ ,,, ƒƒ ,ƒƒƒƒƒƒ,,,,, , ƒƒƒƒ,ˆŽ’Ž,‹,,‰” ,,,

ƒƒƒƒƒƒ□ƒ□ƒƒƒ ,GDIƒƒ□ƒ¥ƒ□ƒ

DirectDraw のの
¥ のの
DirectDraw 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 ¥の
GDI GDI

1 ƒƒƒ¥ƒƒƒƒ,,,,Ž,ƒƒƒƒƒ ƒ ƒƒƒ, ,,

2 Šƒƒƒƒƒ ƒ ƒƒƒ(GDIƒ ƒƒƒ),’Šƒƒƒ¥ƒƒƒƒ,ƒƒƒƒ“‘,,

3 GDI,‘, ,’,, Šƒƒ ,‰Ž‰,,,, *lpDDSurfaceTargetOverride*
ƒƒƒ ƒ,NULL,ƒƒƒ,,ƒ ƒƒƒ,ƒƒƒƒƒƒ,,

GDI ののGDI のの

```
pPrimary->Flip(pMiddle);
```

ƒ□ƒƒƒ,□Ž

¥ の DirectDrawSurface
DirectDrawSurface
DirectDrawSurface
¥の **DDERR_SURFACELOST**

¥の¥ の
IDirectDrawSurface2::Restore
DirectDrawSurface

の

□•Š,ƒƒ□ƒƒƒ•Š

RGB ¥ 4 FOURCC)

IDirectDrawSurface2::GetPixelFormat
RGB DDPF_FOURCC
DDPIXELFORMAT 構造体の **dwFourCC** メンバ FOURCC
YUV DDPF_YUV
dwYUVBitCount **dwYBits** **dwUBits** **dwVBits** **dwYUVAlphaBits** メンバ

RGB DDPF_RGB
dwRGBBitCount **dwRBits** **dwGBits** **dwBBits** **dwRGBAlphaBits**
RGB
DDPF_RGB DDPF_FOURCC

FOURCC
FOURCC のののの

ƒƒ□¥ƒ□ƒƒƒ

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¥の

8

```
// ƒƒƒƒ□ƒƒƒƒ26,ƒƒ□¥ƒ□.
dwColorSpaceLowValue = 26;
dwColorSpaceHighValue = 26;
```

24

```
// □255,128,128,ƒƒ□¥ƒ□
dwColorSpaceLowValue = RGBQUAD(255,128,128);
dwColorSpaceHighValue = RGBQUAD(255,128,128);
```

FourCC YUV

```
// YUVƒƒ□  Y,100,110,Š
//  U,,, V,50,55,Š,‟‰,,,
dwColorSpaceLowValue = YUVQUAD(100,50,50);
dwColorSpaceHighValue = YUVQUAD(110,55,55);
```

ƒ□ƒ□ƒƒZƒ□ƒ□

Z □□□□

DirectDraw ¥ ¥
¥のの
¥のの
¥ Z

Z
の Z
Z 0 Z
0 40
Z 2 Z 1の Z

ƒƒƒƒƒ□ƒ□ƒ□ƒƒƒ,,,,•□ƒƒƒƒ

DirectDraw のののの¥
16 のの

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

ののののの¥の4 8

3D

¥ ¥のの
¥

ƒƒƒƒ“‘ ,ƒƒƒƒ□ƒƒƒ□ƒ□ƒƒƒ

ブロック転送 DMA
¥の **DDCAPS** 構造体
12

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

SVB ¥ ブロック転送
VSB ¥ ブロック転送
SSB ブロック転送

dwSVBCaps メンバ **dwCaps** メンバ
ブロック転送 ¥ の
dwSVBCKeyCaps メンバ **dwCKeyCaps** メンバ **dwSVBFXCaps** メンバ **dwFXCaps** メンバ **dwSVBRops** メンバ ブロック転送

dwCaps **DDCAPS_CANBLTSYSMEM**
のの
ブロック転送

¥ ¥
DirectDraw
IDirectDrawSurface2::PageLock

DirectDraw *ff□ffff*

のの
DirectDraw アプリケーション 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: fffff f f fff,,,ffff“*‘(DDEX3)
- *ff ffff4: 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*ffffffff*C++*,* *,,,,,*
C*fffff,Ž,,,,* *fffff,’,,,,ffff,“‹•* *,,,,,,,,,* C*fffff,’,,,,,* *,,,,*vtable*fff ffff¥ffff,,,,,*this*ffff,‰,,•—*
,,, ,,, ,, •, *C,Ž,,COMffffff,ffff ,Ž ,,,,*

Tutorial 1: DirectDraw,Š‘

DirectDraw
DirectDraw

DirectDrawSurface
SDK
DDEX1 のの

- *Step 1: DirectDrawffffff ,*

- *Step 2: ƒƒƒƒ ƒƒƒ‚ ‚•‚‚Œ’*
- *Step 3: ƒƒƒƒƒƒ ƒ ƒ‚•*
- *Step 4: ƒƒƒƒƒƒ ƒ ƒƒƒ ‚*
- *Step 5: ƒ ƒƒƒ ‚‚ƒƒƒƒƒƒ*
- *Step 6: ƒ ƒƒƒ‚‚ ‚‚‚*
- *Step 7: ƒ ƒƒƒ‚ƒƒƒƒƒƒ*
- *Step 8: DirectDrawƒƒƒƒƒƒ‚Š‚“‚‰•*

Step 1: DirectDrawƒƒƒƒƒƒ‚□□

DirectDraw DDEX1 **doInit**
DirectDrawCreate **DirectDrawCreate** 3

¥ (GUID) GUID NULL
DirectDraw
の DirectDraw
のの NULL
の

の DirectDraw

```
ddrval = DirectDrawCreate(NULL, &lpDD, NULL);
if(ddrval == DD_OK)
{
    // lpDD‚ªŒ‚DirectDrawƒƒƒƒƒƒ‚‚‚
}
else
{
    //  DirectDrawƒƒƒƒƒƒ‚□□‚‚‚‚‚‚
}
```

Step 2: ƒƒƒƒ□ƒƒƒ‚□‚•‚‚Œ’

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)
{
    // ″‵‶ƒ□ƒ,□Œ,,.
}
else
{
    // ″‵‶ƒ□ƒ,Ž″,,
    // ,,,□ƒƒƒƒ□ƒƒƒ,,,‶,,,,
}
```

IDirectDraw::SetCooperativeLevel DD_OK

IDirectDraw::SetCooperativeLevel
Windows ¥ (**HWND**)
(GP) GDI ¥
Windows DirectDraw

DirectDraw
のののの
¥

GDI ALT+TAB

Step 3: ƒƒƒƒƒƒ□ƒ□ƒ,•□

の **IDirectDraw::SetDisplayMode**
640×480×8 bpp

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

ddrval = lpDD->SetDisplayMode(640, 480, 8);
if(ddrval == DD_OK)
{
    // ƒƒƒƒƒƒ¥ƒ□ƒ,•□,□Œ,,
}
else
{
    // ƒƒƒƒƒƒ□ƒ□ƒ,•□,,,,,,
    // ƒ□ƒ,ƒƒ□ƒ,,,,,,,□‵,ƒƒƒƒ□ƒƒƒ,
    // ″‵‶ƒ□ƒ,,,,,,
}
```

ののの640×480×8 のの
(
IDirectDraw::SetDisplayMode DDERR_INVALIDMODE

IDirectDraw::EnumDisplayModes)

Step 4: ƒƒƒƒƒƒ□ƒ□ƒƒƒ,□□

DDEX1

IDirectDraw::SetCooperativeLevel DDSCL_NORMAL **IDirectDraw::SetCooperativeLevel**

- ƒ ƒƒƒ—‹,’‹
- ƒ ƒƒƒ

の

の **DDSURFACEDESC** 構造体の のの

```
// ƒƒƒ¥ƒƒƒƒ,,,,Ž,ƒƒƒƒƒ□ƒ□ƒƒƒ,□□,,
ddsd.dwSize = sizeof(ddsd);
ddsd.dwFlags = DDSD_CAPS | DDSD_BACKBUFFERCOUNT;
ddsd.ddsCaps.dwCaps = DDSCAPS_PRIMARYSURFACE |
    DDSCAPS_FLIP | DDSCAPS_COMPLEX;

ddsd.dwBackBufferCount = 1;
```

の **dwSize** メンバに **DDSURFACEDESC** 構造体の DirectDraw **dwSize** メンバ **DDSURFACEDESC** 構造体ののののの

dwFlags メンバ **DDSURFACEDESC** 構造体のの DDEX1 の **dwFlags** **DDSCAPS** 構造体 (DDSD_CAPS) (DDSD_BACKBUFFERCOUNT)

dwCaps メンバ **DDSCAPS** 構造体 (DDSCAPS_PRIMARYSURFACE) (DDSCAPS_FLIP) (DDSCAPS_COMPLEX)

の ¥ ¥ ¥ DDEX1 のの 1 ¥ ¥ .

¥ DirectDraw ¥ (1MB RAM ¥) **DDSCAPS** 構造体の **dwCaps** メンバ DDSCAPS_SYSTEMMEMORY DDSCAPS_VIDEOMEMORY ¥ DDSCAPS_VIDEOMEMORY **IDirectDraw::CreateSurface** DDERR_OUTOFVIDEOMEMORY

ƒ□ƒƒƒ,□□

DDSURFACEDESC 構造体のメンバを **DirectDrawCreate** DirectDraw *lpDD* **DDSURFACEDESC** 構造体とを **IDirectDraw::CreateSurface**

```
ddrval = lpDD->CreateSurface(&ddsd, &lpDDSPrimary, NULL);
if(ddrval == DD_OK)
{
    // □,,ƒ□ƒƒ,Ž,lpDDSPrimary
}
else
{
    // ƒ□ƒƒ,□□,,,,,,
    return FALSE;
}
```

lpDDSPrimary
IDirectDraw::CreateSurface

のの¥ の
IDirectDrawSurface::GetAttachedSurface

```
ddscaps.dwCaps = DDSCAPS_BACKBUFFER;
ddrval = lpDDSPrimary->GetAttachedSurface(&ddcaps, &lpDDSBack);
if(ddrval == DD_OK)
{
    // lpDDSBack,□ƒƒƒ¥ƒƒƒƒ,,ƒƒƒƒ
}
else
{
    return FALSE;
}
```

ののDDSCAPS_BACKBUFFER
IDirectDrawSurface::GetAttachedSurface *lpDDSBack* ¥

Step 5: *ƒ□ƒƒƒ„ƒƒƒƒƒƒ*

¥ DDEX1
Windows GDI ¥ ¥ のの

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

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

¥ 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**,ƒ ƒƒƒ,ƒƒƒ‰ ,,Œ ƒ ƒƒƒ ƒƒƒ,,ƒƒƒƒ, –Œ,,, ƒ ƒƒƒ ƒƒƒ,,—
Œ,ƒƒƒƒ,“,,, **IDirectDrawSurface::Lock**, ,Ž,,,,,,,,,

Step 6: ƒ□ƒƒƒ,,□,,,

DDEX1 WM_TIMER

```
case WM_TIMER:
    // ƒ□ƒƒƒ,ƒƒƒƒƒƒ
    if(bActive)
    {
        if (lpDDSBack->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;
            }
            lpDDSBack->ReleaseDC(hdc);
        }
```

IDirectDrawSurface2::GetDC
¥ **SetBkColor** **SetTextColor**

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

¥
IDirectDrawSurface::ReleaseDC

Step 7: ƒƒƒƒƒƒ□ƒ□ƒƒƒ

¥
フリッピング **IDirectDrawSurface::Flip**

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

lpDDSPrimary ¥
IDirectDrawSurface::Flip
フリッピング DD_OK while

フリッピング DDERR_SURFACELOST
IDirectDrawSurface::Restore の
IDirectDrawSurface::Flip while

Note **IDirectDrawSurface::Flip**,Œ, , ƒƒƒƒƒƒ,,,,,Š— ,, Ž, ’‹ ŠŠ, ,,, ,,,,‘,ƒƒƒƒƒƒ, ,,,,,,,, **IDirectDrawSurface::Flip**, DDERR_WASSTILLDRAWING,•, —,, **IDirectDrawSurface::Flip**,DD_OK,•,,,ƒ ƒ,‘,,

Step 8: DirectDraw*ƒƒƒƒƒƒ*,Š,“,□□

F12 DDEX1 WM_DESTROY のの **IUnknown::Release** **finiObjects**

```
static void finiObjects(void)
{
    if(lpDD != NULL)
    {
        if(lpDDSPrimary != NULL)
        {
            lpDDSPrimary->Release();
            lpDDSPrimary = NULL;
        }
        lpDD->Release();
        lpDD = NULL;
    }
} // finiObjects
```

DirectDraw (**lpDD**) DirectDrawSurface (**lpDDSPrimary**) NULL DDEX1 DirectDrawSurface 1 **IDirectDrawSurface::Release** 0 DirectDrawSurface DirectDrawSurface NULL DirectDraw 0 **IDirectDraw::Release** DirectDraw の NULL

Tutorial 2: *ƒƒƒƒƒƒ*,*ƒƒƒ*□*ƒƒƒƒ*,“‘

(DDEX2) 1 DDEX1 の DDEX2 ¥のの

- *Step 1: ƒƒƒƒ,*
- *Step 2: ƒƒƒƒ, ’*
- *Step 3: ƒƒƒ ƒƒƒƒ,,ƒƒƒƒƒƒ,ƒ ƒ*
- *Step 4: ƒ ƒƒƒ ƒƒƒƒƒƒ*

DDEX1 **doInit** DDEX2 DirectDraw DDEX1 DDEX2

```
lpDDPal = DDLoadPalette(lpDD, szBackground);
```

```
if (lpDDPal == NULL)
    goto error;

ddrval = lpDDSPrimary->SetPalette(lpDDPal);

if(ddrval != DD_OK)
    goto error;

// ƒƒƒƒƒƒ‚ƒƒƒ□ƒƒƒƒ‚ƒ□ƒ
ddrval = DDReLoadBitmap(lpDDSBack, szBackground);

if(ddrval != DD_OK)
    goto error;
```

Step 1: *ƒƒƒƒ*‚□□

DDEX2 の

```
lpDDPal = DDLoadPalette(lpDD, szBackground);

if (lpDDPal == NULL)
    goto error;
```

DDLoadPalette DirectDraw \Sdk\Samples\Misc\Ddutil.cpp DirectDraw のX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: ƒƒƒƒ‚□’

IDirectDrawSurface::SetPalette
DirectDrawPalette *ddpal* プライマリ・サーフェス
の

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

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

IDirectDrawSurface::SetPalette DirectDrawPalette
DirectDrawSurface

Step 3: ƒƒƒ□ƒƒƒƒ„ƒƒƒƒƒƒ‚ƒ□ƒ

DirectDrawPalette DirectDrawSurface
DDEX2 の Back.bmp ¥

```
// ƒƒƒ□ƒƒƒƒ‚ƒƒƒƒƒƒ‚ƒ□ƒ
ddrval = DDReLoadBitmap(lpDDSBack, szBackground);

if(ddrval != DD_OK)
    // ƒ□ƒ‚Ž”‚‚
```

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

DDCopyBitmap **GetObject** ¥の

```
// ƒ□ƒƒƒ‚ƒƒƒ‚‟‚
ddsd.dwSize = sizeof(ddsd);
ddsd.dwFlags = DDSD_HEIGHT | DDSD_WIDTH;
pdds->GetSurfaceDesc(&ddsd);
```

ddsd **DDSURFACEDESC** 構造体
DirectDraw ののの **DDSURFACEDESC** メ
ンバ DDSD_HEIGHT DDSD_WIDTH の
IDirectDrawSurface::GetSurfaceDesc
DDEX2 480 640

DDCopyBitmap ¥

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

```
{
    StretchBlt(hdc, 0, 0, ddsd.dwWidth, ddsd.dwHeight, hdcImage, x,
y,
        dx, dy, SRCCOPY);
    pdds->ReleaseDC(hdc);
}
```

Step 4: ƒ□ƒƒƒ□ƒƒƒƒƒƒ

DDEX2 のの □□□□□□
DDERR_SURFACELOST DDEX1
1: DirectDraw の) の
DDReLoadBitmap ¥

Tutorial 3: ƒƒƒƒƒ□ƒ□ƒ□ƒƒƒ„ƒƒƒƒ“‘

2 (DDEX2)の¥ ¥
□□□□□□□□□ の

のの
3(DDEX3) の DDEX2 のの

DDEX3

- *Step 1: ƒƒƒƒƒ ƒ ƒ ƒƒƒ‚*
- *Step 2: ƒƒƒƒƒ ƒ ƒ ƒƒƒ‚ƒƒƒƒƒƒ‚ƒ ƒ*
- *Step 3: ƒƒƒ ƒƒƒƒ„ƒƒƒƒƒ ƒ ƒ ƒƒƒ‚ƒƒƒƒ“‘*

Step 1: ƒƒƒƒƒ□ƒ□ƒ□ƒƒƒ‚□□

のの DDEX3 **doInit**

```
// ƒƒƒƒƒ□ƒƒƒƒƒƒƒ‚□□
ddsd.dwFlags = DDSD_CAPS | DDSD_HEIGHT | DDSD_WIDTH;
ddsd.ddsCaps.dwCaps = DDSCAPS_OFFSCREENPLAIN;
ddsd.dwHeight = 480;
ddsd.dwWidth = 640;
ddrval = lpDD->CreateSurface(&ddsd, &lpDDSOne, NULL);
if(ddrval != DD_OK)
{
    return initFail(hwnd);
}

// ‚‚‚‚‚‚ƒƒƒƒƒ□ƒƒƒƒƒƒƒ‚□□
ddrval = lpDD->CreateSurface(&ddsd, &lpDDSTwo, NULL);
if(ddrval != DD_OK)
{
    return initFail(hwnd);
}
```

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

¥ ¥ ¥
IDirectDraw::CreateSurface

DDSCAPS □□□の DDSCAPS_SYSTEMMEMORY
DDSCAPS_VIDEOMEMORY ¥

¥
¥の¥
¥ DDSCAPS_VIDEOMEMORY
のDDERR_OUTOFVIDEOMEMORY

Step 2: *ƒƒƒƒƒ□ƒ□ƒ□ƒƒƒ‚ƒƒƒƒƒƒ‚ƒ□ƒ*

の DDEX3 Frntback.bmp
InitSurfaces
InitSurfaces Ddutil.cpp
DDCopyBitmap

```
// ƒƒƒƒƒƒƒƒ□ƒ,ƒ□ƒ
hbm = (HBITMAP)LoadImage(GetModuleHandle(NULL), szBitmap,
    IMAGE_BITMAP, 0, 0, LR_CREATEDIBSECTION);

if (hbm == NULL)
    return FALSE;

DDCopyBitmap(lpDDSOne, hbm, 0, 0,   640, 480);
DDCopyBitmap(lpDDSTwo, hbm, 0, 480, 640, 480);
DeleteObject(hbm);

return TRUE;
```

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

Step 3: *ƒƒƒ□ƒƒƒƒ„ƒƒƒƒƒ□ƒ□ƒ□ƒƒƒ‚ƒƒƒƒ“‘*

WM_TIMER
DDEX3
¥

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

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

phase ¥ ブロック転送 **IDirectDrawSurface::BltFast** ¥の(0,0) ブロック転送 *rcRect* **RECT** 構造体であり、アンパサンド（&）をつけてそ の のの の FALSE (0)

IDirectDrawSurface::Blt **IDirectDrawSurface::BltFast** の¥ ¥ ¥ のの **IDirectDrawSurface::BltFast**

10 ¥ ¥ **IDirectDrawSurface::BltFast** **IDirectDrawSurface::Blt**

¥ ¥のの フリッピング

Tutorial 4: ƒƒ□¥ƒ□,ƒƒƒƒƒƒƒƒƒ□ƒƒƒ

(DDEX3)の¥の (DDEX4) ののの の

DDEX4 All.bmp DDEX4 のの¥ ¥

DDEX4

- *Step 1: ƒƒ ¥ƒ , ’*

- *Step 2: Š’‚ƒƒƒ ƒƒƒ‚*

Step 1: *ƒƒ*□*¥ƒ*□‚□’

DirectDraw の **doInit** DDEX4 のの
の□□□¥□□ □□□¥□□
□□□¥□□
の

```
//□‚ƒƒƒƒƒ‚ƒƒ□¥ƒ□‚□’‚‚
DDSetColorKey(lpDDSOne, RGB(0,0,0));

return TRUE;
```

DDSetColorKey RGB ¥ の
RGB (0, 0, 0) **DDSetColorKey** **DDColorMatch**
Ddutil.cpp **DDColorMatch** *lpDDSOne*
(0, 0)ののの RGB (0, 0)
のののの(0, 0) ¥ の
DDSetColorKey **DDColorMatch** ¥
DDCOLORKEY □□□の **dwColorSpaceLowValue** □□□
dwColorSpaceHighValue □□□
IDirectDrawSurface::SetColorKey ¥

DDSetColorKey **DDColorMatch** CLR_INVALID
DDEX4 **DDSetColorKey** ¥
CLR_INVALID (0, 0)
¥ DDEX4 (0, 0)
の DDEX4 の (0, 0)
¥ All.bmp
(0, 0)

```
DDSetColorKey(lpDDSOne, CLR_INVALID);
```

DDEX4 の Ddex4.rc の
DDEX4 の¥ (0, 0)

Step 2: Š’‚*ƒƒƒ*□*ƒƒƒ*‚□□

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

IDirectDrawSurface::BltFast
dwTrans
DDBLTFAST_NOCOLORKEY
¥ *dwTrans*
DDBLTFAST_SRCCOLORKEY
¥ の *lpDDSOne* □□□□□□

の **updateFrame** の

DDEX4 のの

Tutorial 5: ƒƒƒƒ‚““•□

(DDEX5) 4 (DDEX4) の

DDEX5

- *Step 1: ƒƒƒƒ ƒƒƒƒ‚ƒ ƒ*
- *Step 2: ƒƒƒƒ‚ƒ ƒ ƒ*

Step 1: ƒƒƒƒ□ƒƒƒƒ‚ƒ□ƒ

DDEX5 のの All.bmp のの の

```
// ‚‚‚‚□‚‚‚‚□‚–Ž—‚ƒƒƒ‚‚
for(i=0; i<256; i++)
{
    torusColors[i] = 0;
}

// ƒ□ƒƒƒ‚ƒƒƒ‚□Œ”••‚ƒƒƒƒ‚‚□ƒ□ƒƒ¥ƒƒƒ□
// Œ‚‚‚‚ƒƒƒƒƒƒ‚‹˜‚•Ž‚‚
ddsd.dwSize = sizeof(ddsd);
while (lpDDSOne->Lock(NULL, &ddsd, 0, NULL) == DDERR_WASSTILLDRAWING)
    ;

// ƒ□ƒƒ¥ƒƒ□ƒ‚Œ□‚□Ž—□‚ƒ□ƒ‚‚
for(y=480; y<480+384; y++)
{
    for(x=0; x<640; x++)
    {
        torusColors[((BYTE *)ddsd.lpSurface)[y*ddsd.lPitch+x]] = 1;
    }
}

lpDDSOne->Unlock(NULL);
```

torusColors All.bmp のの
torusColors 0

torusColors 480 0 の¥ のの
DDSURFACEDESC の **lpSurface** メンバ
480 0 (y × **lPitch** + x)
1 y (row) **lPitch**
DDSURFACEDESC のののの

torusColors のの

true "*ddsd.lPitch"
(y×**lPitch**

)

Step 2: ƒƒƒƒ‚ƒ□ƒ□ƒ

DDEX5 **updateFrame** Tutorial 4 (DDEX4)
¥ 3
フリッピング **updateFrame** の **doInit**

```
// ƒƒƒƒ‚•□
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 フリッピング

‚‚‘‚ DirectDrawƒƒƒƒ

DirectDraw DirectX SDK のの

- Stretch

  ”””““ ƒƒƒƒƒ ƒ ƒ¥ƒƒƒ ƒƒƒ‚ ƒƒƒƒƒƒ‚‚‚“‘‚ ’‚ƒƒƒƒƒƒ‚‚‚“‘‚”—
  ‚‚‚ ƒƒƒƒƒ“‚ ‚‚•–‚Ž‚
- Donut

  ‘‚‚”““ƒ ƒ ƒƒƒƒ ƒƒƒ‚”””““ƒ ƒ ƒƒƒƒ ƒƒƒ‚‘Œ —‚‚Ž‚‚Ž‚
- Wormhole

  ƒƒƒƒ ƒƒƒ ƒƒƒ‚•–‚Ž‚
- Dxview

  ƒƒƒƒƒƒ¥ƒ ƒƒƒƒ‚”—‚’‚‚•–‚Ž‚

DirectDraw Duel, Iklowns, Foxbear, Palette,Flip2d

□“‰‚ƒƒƒƒƒƒ

SDK DirectDraw のののの

- ƒƒƒƒƒƒ‚ƒƒƒƒ“‘‚ ‘Ž“
- ƒƒ ƒƒƒ‚”‚ƒƒƒƒ“‘
- ƒƒƒƒƒƒ¥ƒ ƒƒƒƒ”—‚Œ’
- ƒƒƒƒƒƒ¥ƒƒƒ‚‚ƒƒƒƒƒƒ‚Š”
- ƒƒƒƒ ƒƒƒƒƒƒƒ

ƒƒƒƒƒƒ‚ƒƒƒƒ“‘‚□‘Ž“

IDirectDrawSurface2::Flip プライマリ・サーフェス
¥ のの フリッピン
グ
DDERR_WASSTILLDRAWING SDK の **IDirectDrawSurface2::Flip**
DD_OK
IDirectDrawSurface2::Flip

DDERR_WASSTILLDRAWING
フリッピング ¥
IDirectDrawSurface2::GetFlipStatus

のフリッピング DDERR_WASSTILLDRAWING

フリッピング

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

    // ‘,ƒƒƒƒƒƒ,□–,,,,‘,
    // ƒƒƒƒ□ƒƒƒ,,,,□‘,ƒƒƒ,Ž□,,,

ddrval = lpDDSPrimary->Flip(NULL, 0);
```

ブロック転送

IDirectDrawSurface2::GetBltStatus

IDirectDrawSurface2::GetFlipStatus **IDirectDrawSurface2::GetBltStatus**

ƒƒ□¥ƒƒƒ,”,ƒƒƒƒ“‘

¥ **IDirectDrawSurface2::Blt**

DDBLT_COLORFILL

IDirectDrawSurface2::Blt

の¥の

```
DDBLTFX ddbltfx;

ddbltfx.dwSize = sizeof(ddbltfx);
ddbltfx.dwFillColor = 0;
ddrval = lpDDSPrimary->Blt(
    NULL,        // “‘□
    NULL, NULL,  // “‘Œ‹Œ
    DDBLT_COLORFILL, &ddbltfx);

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

ƒƒƒƒƒƒ¥ƒ□ƒƒƒƒ”—‚Œ’

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

¥ **IDirectDraw2::GetCaps**
DirectDraw **DDCAPS** 構造
体の **dwCaps** メンバ ののの
DDCAPS 構造体 の
DDSCAPS 構造体の DirectDraw のの **DDSCAPS** 構造体
.

ƒƒƒƒƒƒ¥ƒƒƒ„ƒƒƒƒƒƒ‚Š”

¥ ¥
¥のの¥

¥ ¥ののの
DDCAPS 構造体の **dwVidMemTotal** **dwVidMemFree**
(の¥ **IDirectDraw2::GetCaps**
) の DirectX SDK DirectX
¥ DirectDraw の ¥
¥ ¥ ¥
のの
DirectDraw

ƒƒƒƒ¥ƒƒƒƒƒƒƒ

ののの¥の¥ ¥

```
// ‚‚‚‚ƒƒƒ¥ƒƒƒƒ‚ƒƒƒƒƒ□ƒ□ƒƒƒ‚□□
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, &lpDDSPrimary, NULL);
if(ddrval == DD_OK)
{
    // ‘ˆ‚ƒƒƒ¥ƒƒƒƒ‚‚ƒƒƒƒ‚Ž“
    ddscaps.dwCaps = DDSCAPS_BACKBUFFER;
    ddrval = lpDDSPrimary->GetAttachedSurface(&ddscaps,
        &lpDDSBackOne);
    if(ddrval != DD_OK)
        // ‚‚‚ƒƒ□ƒƒƒ□ƒ‚•Ž
    // ‘“‚ƒƒƒ¥ƒƒƒƒ‚‚ƒƒƒƒ‚Ž“
    ddscaps.dwCaps = DDSCAPS_BACKBUFFER;
    ddrval = lpDDSPrimary->GetAttachedSurface(&ddscaps,
        &lpDDSBackTwo);
```

フリッピング ¥
¥
フリッピング
¥ **IDirectDrawSurface2::Flip** DD_OK

DirectDrawƒƒƒƒƒ

Š□

DirectDrawCreate

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

DirectDraw

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_DIRECTDRAWALREADYCREATED
  DDERR_GENERIC
  DDERR_INVALIDDIRECTDRAWGUID
  DDERR_INVALIDPARAMS
  DDERR_NODIRECTDRAWHW
  DDERR_OUTOFMEMORY

lpGUID
,,,ƒƒƒƒ,•,ƒƒ ƒƒƒƒ ƒŽ•Ž (GUID),,ƒƒƒƒ
NULL,,,, ƒƒƒƒƒ,ƒƒƒƒƒƒ ƒƒƒƒ,Ž,

lplpDD
Œ, ,, Œ,, —Œ,DirectDrawƒƒƒƒƒƒ,ƒƒƒƒ, Š‰,,,ƒƒƒƒ,,ƒƒƒƒ

pUnkOuter
COM ‰‹”, —,ŒŠ ,,,,,, Œ , NULL, ’,,,,ƒƒ ,,,

DirectDraw *lplpDD* **IDirectDraw2::GetCaps**

DirectDrawCreateClipper

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

DirectDraw DirectDrawClipper

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDPARAMS

  DDERR_OUTOFMEMORY

dwFlags
,,ƒƒƒ ƒ,Œ –Ž—, 0,ƒƒƒ,,,,,,,,,,,

lplpDDClipper
,,DirectDrawClipperƒƒƒƒƒƒ,,ƒƒƒƒ,•,,,ƒƒƒƒ,,ƒƒƒƒ

pUnkOuter
COM ‰‹”, —,ŒŠ ,,,,,, Œ , NULL, ’,,,,ƒƒ ,,,

DirectDraw
DirectDrawClipper DirectDraw DirectDrawClipper
DirectDraw

DirectDraw DirectDrawClipper
IDirectDraw2::CreateClipper

IDirectDraw2::CreateClipper

DirectDrawEnumerate

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

DirectDraw NULL GUID
GDI

- Œ,,,DD_OK Ž”,,,**DDERR_INVALIDPARAMS**,•,

lpCallback
ƒƒƒƒ,HALƒƒƒƒƒƒ,ƒƒƒ ƒ,,,Œ, ,,, **Callback**Š ,ƒƒƒƒ

lpContext
—‹ƒ ƒƒƒƒŠ ,Œ, ,,,,,,“,,,ƒƒƒƒ ƒƒƒ,’‹,,ƒƒƒƒƒƒ,,ƒƒƒƒ

ƒ□ƒƒƒƒŠ□

Callback

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

DirectDrawEnumerate

- —‹,‘,, ,DDENUMRET_OK,•, Ž,, ,DDENUMRET_CANCEL,•,

lpGUID
DirectDrawƒƒƒƒƒƒ“,ƒƒ ƒƒƒƒ ƒŽ•Ž ,,,, ,ƒƒƒƒ

lpDriverDescription
ƒƒƒƒ‚Š‚‚‹ ‚Ž‚•Ž—‚ƒƒƒƒ

lpDriverName
ƒƒƒƒ–‚Ž‚•Ž—‚ƒƒƒƒ

lpContext
Š ‚Œ‚ ‚‚‚‚‚‚ƒ ƒƒƒƒŠ ‚“‚‚‚ƒƒƒƒ ƒƒƒ‚’‹‚‚ ‘‘‚‚ƒƒƒƒ

EnumModesCallback

```
HRESULT WINAPI lpEnumModesCallback(LPDDSURFACEDESC lpDDSurfaceDesc,
    LPVOID lpContext);
```

IDirectDraw2::EnumDisplayModes のの

- —‹‚‘‚‚ ‚DDENUMRET_OK‚•‚ Ž‚‚ ‚DDENUMRET_CANCEL‚•‚

lpDDSurfaceDesc
ƒƒƒ ‚Ž” ‚ ‚‚“‚ƒ ƒ‚’‹‚‚**DDSURFACEDESC** ‘‘ ‚ƒƒƒƒ ‚‚ƒ ƒ‚“‚ ‚ —
‚‚‚

lpContext
Š ‚Œ‚ ‚‚‚‚‚‚ƒ ƒƒƒƒŠ ‚“‚‚‚ƒƒƒƒ ƒƒƒ‚’‹‚‚ ‘‘‚‚ƒƒƒƒ

EnumSurfacesCallback

```
HRESULT WINAPI lpEnumSurfacesCallback(
    LPDIRECTDRAWSURFACE2 lpDDSurface,
    LPDDSURFACEDESC lpDDSurfaceDesc, LPVOID lpContext);
```

IDirectDrawSurface2::EnumAttachedSurfaces のの

- —‹‚‘‚‚ ‚ DDENUMRET_OK‚•‚ Ž‚‚ ‚DDENUMRET_CANCEL‚•‚

lpDDSurface
‚‚ƒ ƒƒƒ‚ƒƒƒƒ‚‚‚ƒ ƒƒƒ‚ƒƒƒƒ

lpDDSurfaceDesc
ƒƒƒƒ‚‚‚ƒ ƒƒƒ‚‹ ‚‚**DDSURFACEDESC** ‘‘‚ƒƒƒƒ

lpContext
Š ‚Œ‚ ‚‚‚‚‚‚ƒ ƒƒƒƒŠ ‚“‚‚‚ƒƒƒƒ ƒƒƒ‚’‹‚‚ ‘‘‚‚ƒƒƒƒ

fnCallback

```
HRESULT WINAPI lpfnCallback(LPDIRECTDRAWSURFACE lpDDSurface,
    LPVOID lpContext);
```

IDirectDrawSurface2::EnumOverlayZOrders のの

- —‹‚‘‚‚ ‚ DDENUMRET_OK‚•‚ Ž‚‚ ‚DDENUMRET_CANCEL‚•‚

lpDDSurface
ƒ ƒƒƒ‚ƒ ƒ ƒƒ‚‚‚‚‚ƒ ƒƒƒ‚ƒƒƒƒ

lpContext
Š ‚Œ‚ ‚‚‚‚‚‚ƒ ƒƒƒƒŠ ‚“‚‚‚ƒƒƒƒ ƒƒƒ‚’‹‚‚ ‘‘‚‚ƒƒƒƒ

IDirectDraw2

DirectDraw
IDirectDraw2
IDirectDraw2
DirectDraw

IDirectDraw2

ƒƒƒ‚Š‚“‚	**Compact** **Initialize**
ƒƒƒƒƒƒ‚	**CreateClipper** **CreatePalette** **CreateSurface**
ƒƒƒƒ‚”—	**GetCaps**
ƒƒƒƒƒƒ¥ƒ ƒ	**EnumDisplayModes** **GetDisplayMode** **GetMonitorFrequency** **RestoreDisplayMode** **SetDisplayMode**
ƒƒƒƒƒƒ¥ƒƒ ƒƒ	**GetScanLine** **GetVerticalBlankStatus**
‚‚‘	**GetAvailableVidMem** **GetFourCCCodes** **WaitForVerticalBlank**
“ Š‹‚ ’	**SetCooperativeLevel**
ƒ ƒƒƒ	**DuplicateSurface** **EnumSurfaces** **FlipToGDISurface** **GetGDISurface**

IDirectDraw2 COM
IUnknown **IUnknown**
3 の

AddRef

QueryInterface

Release

IDirectDraw2::Compact

```
HRESULT Compact();
```

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_NOEXCLUSIVEMODE

  DDERR_SURFACEBUSY

¥ ののの

IDirectDraw2::CreateClipper

```
HRESULT CreateClipper(DWORD dwFlags,
    LPDIRECTDRAWCLIPPER FAR * lplpDDClipper,
    IUnknown FAR * pUnkOuter);
```

DirectDrawClipper

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_NOCOOPERATIVELEVELSET

  DDERR_OUTOFMEMORY

dwFlags
,,ƒƒƒ ƒ,Œ –Ž—, 0,ƒƒƒ,,,,,,,,,,,

lplpDDClipper
,,ƒƒƒƒ, Œ,, •,,, ,,DirectDrawClipperƒƒƒƒƒƒ,Ž,ƒƒƒƒ,ƒƒƒƒ

pUnkOuter
COM ‰‹”, —,ŒŠ ,,,,,, Œ , NULL, ’,,,,ƒƒ ,,,

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

DirectDraw DirectDrawClipper **DirectDrawCreateClipper**

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

IDirectDraw2::CreatePalette

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

DirectDrawPalette

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_NOCOOPERATIVELEVELSET
  DDERR_NOEXCLUSIVEMODE
  DDERR_OUTOFCAPS
  DDERR_OUTOFMEMORY
  DDERR_UNSUPPORTED

dwFlags

Ž,ƒƒƒ,,, ,,,ˆ ,’,Ž,

DDPCAPS_1BIT

ƒƒƒƒƒƒ,1ƒƒƒ ƒƒ ƒ ƒƒ,,ƒƒƒƒ,,,,,,,

DDPCAPS_2BIT

ƒƒƒƒƒƒ,,ƒƒƒ ƒƒ ƒ ƒƒ,,ƒƒƒƒ,,,,,,

DDPCAPS_4BIT

ƒƒƒƒƒƒ,,ƒƒƒ ƒƒ ƒ ƒƒ,,ƒƒƒƒ,,,Œ,,,

DDPCAPS_8BITENTRIES

8ƒƒƒƒƒ ƒƒƒƒƒƒ,Ž ,,ƒƒƒƒƒƒ,Ž, ,,ƒƒƒ, DDPCAPS_1BIT DDPCAPS_2BIT DDPCAPS_4BIT,Ž, ,ƒ ƒƒƒ ƒ ƒƒƒ,8bpp, ,,—
Œ,,, Šƒƒ ƒƒƒƒ, ,ƒƒƒ’, 8bppƒƒƒƒ,ƒ ƒƒƒ,ƒƒƒƒƒƒ,Ž,

DDPCAPS_8BIT

ƒƒƒƒƒƒ,8ƒƒƒ ƒƒ ƒ ƒƒ,,ƒƒƒƒ,256Œ,,,

DDPCAPS_ALLOW256

,,ƒƒƒƒ,256Œ,ƒƒƒƒ,,,,’‹,,,

lpColorTable
,,DirectDrawPalette*ƒƒƒƒƒƒ*, Š‰,,,,,2 4 16
256 ,,,**PALETTEENTRY**Œ”—,*ƒƒƒƒ*

lplpDDPalette
,,*ƒƒƒƒ*, Œ,, •,,, ,,DirectDrawPalette*ƒƒƒƒƒƒ*,,*ƒƒƒƒ*,*ƒƒƒƒ*

pUnkOuter
COM ‰‹”, —,ŒŠ ,,,,,, Œ , NULL, ’,,,,*ƒƒ* ,,,

IDirectDraw2::CreateSurface

```
HRESULT CreateSurface(LPDDSURFACEDESC lpDDSurfaceDesc,
    LPDIRECTDRAWSURFACE FAR * lplpDDSurface,
    IUnknown FAR * pUnkOuter);
```

DirectDrawSurface

- Œ,,, DD_OK Ž”,,,Ž,*ƒƒ* ,•,

  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
,,,*ƒ* *ƒƒƒ*,,—‹,‹ ,,**DDSURFACEDESC** ‘,,*ƒƒƒƒ*

lplpDDSurface
Œ, ,, Œ,, —Œ,DirectDrawSurface*ƒƒƒƒ*,,,, Š‰
,,,*ƒ* *ƒƒƒ*,,*ƒƒƒƒ*,*ƒƒƒƒ*

pUnkOuter
COM ‰‹”, —,ŒŠ ,,,,,, Œ , NULL, ’,,,,*ƒƒ* ,,,

IDirectDraw2::DuplicateSurface

```
HRESULT DuplicateSurface(LPDIRECTDRAWSURFACE lpDDSurface,
    LPLPDIRECTDRAWSURFACE FAR * lplpDupDDSurface);
```

DirectDrawSurface

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_CANTDUPLICATE

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_OUTOFMEMORY

  DDERR_SURFACELOST

lpDDSurface
- ,,DirectDrawSurface ‘‘,ƒƒƒƒ

lplpDupDDSurface
,, ,,,• ,DirectDrawSurface ‘‘,Ž,DirectDrawSurfaceƒƒƒƒ ,ƒƒƒƒ

DirectDrawSurface DirectDrawSurface

のの

3D

IDirectDraw2::EnumDisplayModes

```
HRESULT EnumDisplayModes(DWORD dwFlags,
    LPDDSURFACEDESC lpDDSurfaceDesc, LPVOID lpContext,
    LPDDENUMMODESCALLBACK lpEnumModesCallback);
```

DirectDraw

NULL

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

dwFlags

DDEDM_REFRESHRATES

ˆ,,ƒƒƒƒƒƒƒ ƒ,—‹,,
IDirectDraw2::EnumDisplayModes, Šƒ ƒ,ˆ“,,,,—‹,,,,,,,• ,, ,,ƒƒƒ,
ƒ ƒ,ƒƒ ƒ,,,,,Œ’,,Ž, ƒƒƒƒƒƒƒ ƒ, —,“,,,,,,,,,Ž’,,

lpDDSurfaceDesc

——

,,,ƒ ƒ,ƒƒƒƒ,,**DDSURFACEDESC** ‘‘,,ƒƒƒƒ,ƒƒƒƒ NULL,Ž’,, ,,,,ƒ
ƒ,—‹,,,

lpContext
—‹,,,ƒƒƒ,,,,,“,,,ƒƒƒƒ ƒƒƒ,’‹,, ‘‘,,ƒƒƒƒ

lpEnumModesCallback

Œ, ’,,,,,Œ,,,“,Œ,,,**EnumModesCallback**Š ,,ƒƒƒƒ

DDSURFACEDESC 構造体の **dwRefreshRate** メンバ **IDirectDraw::EnumDisplayModes** **IDirectDraw2::SetDisplayMode** **dwRefreshRate** メンバ **IDirectDraw2::EnumDisplayModes**

IDirectDraw2::GetDisplayMode **IDirectDraw2::SetDisplayMode** **IDirectDraw2::RestoreDisplayMode**

IDirectDraw2::EnumSurfaces

```
HRESULT EnumSurfaces(DWORD dwFlags, LPDDSURFACEDESC lpDDSD,
    LPVOID lpContext, LPDDENUMSURFACESCALLBACK
lpEnumSurfacesCallback);
```

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

dwFlags

Ž,ƒƒƒ,,,,,,,

DDENUMSURFACES_ALL

Š , ’,,ƒ ƒƒƒ,,,,—‹,,

DDENUMSURFACES_CANBECREATED

Š , ’,, ,, , ,,,ƒ ƒƒƒ,—‹,,

DDENUMSURFACES_DOESEXIST

Š , ’,, ,,,‘ ,,,,ƒ ƒƒƒ,—‹,,

DDENUMSURFACES_MATCH

ƒ ƒƒƒ ƒƒƒƒƒƒ,Š“,,ƒ ƒƒƒ,’,

DDENUMSURFACES_NOMATCH

ƒ ƒƒƒ ƒƒƒƒƒƒ,Š“,,,,,ƒ ƒƒƒ,’,

lpDDSD

—,,,,ƒ ƒƒƒ,’‹,,**DDSURFACEDESC** ‘‘,,ƒƒƒƒ

lpContext

—‹,,,ƒƒƒ,“,,,ƒƒƒƒ ƒƒƒ,’‹,, ‘‘,,ƒƒƒƒ

lpEnumSurfacesCallback

ˆ’, ’,,,,,Œ,,,“,Œ,,,**EnumSurfacesCallback**Š ,,ƒƒƒƒ

DDENUMSURFACES_CANBECREATED

IDirectDraw *dwFlags* の

IDirectDraw2::FlipToGDISurface

```
HRESULT FlipToGDISurface();
```

GDI

- Œ,,,DD_OK,•, Ž”,,,Ž,ƒƒ ƒ ƒ,•,
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_NOTFOUND

GDI ¥
¥ ¥

IDirectDraw2::GetGDISurface

IDirectDraw2::GetAvailableVidMem

```
HRESULT GetAvailableVidMem(LPDDSCAPS lpDDSCaps,
    LPDWORD lpdwTotal, LPDWORD lpdwFree);
```

¥ ののの

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_INVALIDCAPS
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_NODIRECTDRAWHW

lpDDSCaps
ƒ ƒƒƒ, ƒ ƒƒƒƒ”—,Š,**DDSCAPS** ‘‘,,ƒƒƒƒ

lpdwTotal
——,,,ƒƒƒƒƒƒ¥ƒƒƒ,‘——,Ž“,,• ,,ƒƒƒƒ

lpdwFree
Œ ,ƒƒƒƒƒƒƒƒ,‹,——,Ž“,,• ,,ƒƒƒƒ

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))
.
.
.
```

¥
¥ののののの¥ ¥ の
Zの¥ののの
Zの¥ の

IDirectDraw2 **IDirectDraw**

IDirectDraw2::GetCaps

`HRESULT GetCaps(LPDDCAPS lpDDDriverCaps, LPDDCAPS lpDDHELCaps);`

HEL

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS

lpDDDriverCaps
ƒƒƒƒ ƒƒƒƒ,• ,,,ƒ ƒƒƒƒ,”—,Ž“,,**DDCAPS** ‘‘,,ƒƒƒƒ

lpDDHELCaps
HEL,”—,Ž“,,**DDCAPS** ‘‘,,ƒƒƒƒ

DDCAPS

IDirectDraw2::GetDisplayMode

`HRESULT GetDisplayMode(LPDDSURFACEDESC lpDDSurfaceDesc);`

- Œ,,,DD_OK,•, Ž”,,,Ž,,,,ƒƒ ,•,
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_UNSUPPORTEDMODE

lpDDSurfaceDesc
ƒ ƒƒƒ‚Ž‚**DDSURFACEDESC** ‘‘‚‚ƒƒƒƒ

IDirectDraw2::RestoreDisplayMode

IDirectDraw2::SetDisplayMode

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

IDirectDraw2::GetFourCCCodes

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

DirectDraw FourCC のの

- Œ‚‚‚ DD_OK Ž”‚‚‚Ž‚ƒƒ ‚•‚
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS

lpNumCodes
lpCodes‚Ž‚‚‚”—‚ƒƒƒƒ ƒƒƒƒ ‚ ‚‚‚‚‚‚‚‚ƒ ƒ‚Ž”‚‚‚‚ ‚‚
lpNumCodes‚‚—‹‚‚‚’‚ƒƒƒ‚‚ ‚‚ ‚‚‚*lpCodes*‚“‚

lpCodes
FourCCƒ ƒ‚“‚”—‚‚ƒƒƒƒ NULL‚“‚‚‚‚
lpNumCodes‚‚ FourCCƒ ƒ‚ƒƒ ƒ‚‚’‚ƒƒƒ‚‚ ƒƒ ƒ‚‚

IDirectDraw2::GetGDISurface

```
HRESULT GetGDISurface(LPDIRECTDRAWSURFACE FAR * lplpGDIDDSSurface);
```

GDI DirectDrawSurface

- Œ‚‚‚ DD_OK Ž”‚‚‚Ž‚ƒƒ ‚•‚
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_NOTFOUND

lplpGDIDDSSurface
GDI‚ƒƒƒƒƒ ƒ ƒƒƒ‚ƒƒƒƒ ƒ‚‚‚‚DirectDrawSurfaceƒƒƒƒƒƒ‚‚ƒƒƒƒ

IDirectDraw2::FlipToGDISurface

IDirectDraw2::GetMonitorFrequency

```
HRESULT GetMonitorFrequency(LPDWORD lpdwFrequency);
```

DirectDraw 100 60Hz 6000

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_UNSUPPORTED

lpdwFrequency
ƒƒƒ Ž” ,Ž,• ,ƒƒƒƒ

IDirectDraw2::GetScanLine

```
HRESULT GetScanLine(LPDWORD lpdwScanLine);
```

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_UNSUPPORTED
  DDERR_VERTICALBLANKINPROGRESS

lpdwScanLine
ƒƒƒƒƒƒ,Œ •‰,,,,ƒƒƒƒƒƒƒ,Ž“,,• ,ƒƒƒƒ

IDirectDraw2::GetVerticalBlankStatus
IDirectDraw2::WaitForVerticalBlank

IDirectDraw2::GetVerticalBlankStatus

```
HRESULT GetVerticalBlankStatus(LPBOOL lpbIsInVB);
```

の

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS

lpbIsInVB
’‹ , ‘,Ž“,,• ,ƒƒƒƒ ,,ƒƒƒ ƒ, ’‹ ŠŠ,,, , TRUE ,,,,, , FALSE ,,,

IDirectDraw2::WaitForVerticalBlank

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

IDirectDraw2::Initialize

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

CoCreateInstance OLE DirectDraw

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_ALREADYINITIALIZED
  DDERR_DIRECTDRAWALREADYCREATED
  DDERR_GENERIC
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_NODIRECTDRAWHW
  DDERR_NODIRECTDRAWSUPPORT
  DDERR_OUTOFMEMORY

lpGUID
ƒƒƒ ƒƒƒƒŽ•Ž,,,Ž—,,,GUID ƒƒ ƒƒƒƒ ƒŽ•Ž ,ƒƒƒƒ
COM ¥ ¥
DirectDraw **DirectDrawCreate**
DDERR_ALREADYINITIALIZED **IDirectDraw2::Initialize**
DirectDraw **CoCreateInstance**
DDERR_NOTINITIALIZED

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

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

IUnknown::Release

IDirectDraw2::RestoreDisplayMode

```
HRESULT RestoreDisplayMode();
```

の¥ **IDirectDraw2::SetDisplayMode**

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_GENERIC
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_LOCKEDSURFACES
  DDERR_NOEXCLUSIVEMODE

IDirectDraw2::SetDisplayMode

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

IDirectDraw2::SetCooperativeLevel

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

の

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_EXCLUSIVEMODEALREADYSET

  DDERR_HWNDALREADYSET

  DDERR_HWNDSUBCLASSED

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_OUTOFMEMORY

hWnd

ƒƒƒƒ ƒƒƒ,Ž,,,ƒƒƒƒƒ¥ƒƒƒƒ

dwFlags

Ž,ƒƒƒ,’,,, ,,,ˆ ,Ž,

DDSCL_ALLOWMODEX

Mode Xƒƒƒƒƒƒƒ ƒ,Ž—,‹‰

DDSCL_ALLOWREBOOT

”‘“ ƒƒƒƒƒ ƒ ƒ ƒŽ, CTRL+ALT+DEL,‹”,‹‰

DDSCL_EXCLUSIVE

”‘“ƒƒƒ,—‹

DDSCL_FULLSCREEN

”‘“ƒ ƒ, —Ž,‘ƒƒƒƒƒ ƒ ƒƒƒ,‘,,‰“, ,,,,Ž, GDI,–Ž,,,,,

DDSCL_NORMAL

ƒƒƒƒ ƒƒƒ,’ ,Windowsƒƒƒƒ ƒƒƒ,,,‹”,,,,,Ž,

DDSCL_NOWINDOWCHANGES

ƒƒƒƒƒ ƒŽ DirectDraw,ƒƒƒƒ ƒƒƒƒƒƒƒƒ, ‰,,,ƒƒƒƒ,,,,,,,

DDSCL_EXCLUSIVE **IDirectDraw2::Compact**

DDSCL_EXCLUSIVE DDSCL_FULLSCREEN
IDirectDraw2::SetCooperativeLevel
の DDSCL_EXCLUSIVE
DDSCL_NORMAL
DDSCL_EXCLUSIVE DDSCL_FULLSCREEN の

X DDSCL_ALLOWMODEX
DDSCL_FULLSCREEN DDSCL_EXCLUSIVE
DDSCL_ALLOWMODEX DDSCL_NORMAL
DDSCL_ALLOWMODEX
IDirectDraw2::EnumDisplayModes X

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

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

IDirectDraw2::SetCooperativeLevel DirectDraw
DDSCL_NORMAL
¥ ¥ NULL
Windows DirectDraw

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

1 ”‘“ƒ ƒ,,,,,, DDSCL_EXCLUSIVEƒƒƒ,ƒƒƒ,, **IDirectDraw::SetCooperativeLevel**,Œ, ,

2 ƒƒƒƒƒƒƒ ƒ,• ,,,,, **IDirectDraw::SetDisplayMode**,Œ, ,

3
”‘“ƒ ƒ,‰ ,,,,,DDSCL_NORMALƒƒƒ,‹,**IDirectDraw::SetCooperativeLevel**,Œ, ,

IDirectDraw2

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

IDirectDraw2::SetDisplayMode

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

¥

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_GENERIC
  DDERR_INVALIDMODE
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_LOCKEDSURFACES
  DDERR_NOEXCLUSIVEMODE

DDERR_SURFACEBUSY

DDERR_UNSUPPORTED

DDERR_UNSUPPORTEDMODE

DDERR_WASSTILLDRAWING

dwWidth and *dwHeight*

,,ƒ ƒ,•, ,

dwBPP

,,ƒ ƒ,ƒƒƒ ƒƒƒƒ(bpp)

dwRefreshRate

,,ƒ ƒ,ƒƒƒƒƒƒƒ ƒ ,,ƒƒƒ ƒ,,,ƒƒƒ,,,, ,,ƒƒƒƒ,**IDirectDraw** ƒƒƒ ƒƒƒƒ¥ƒ ƒƒƒ,Ž—,,,

dwFlags

,,ƒƒƒ ƒ,Œ –Ž— 0,ƒƒƒ,,,,,,

IDirectDraw2::SetCooperativeLevel

DirectDrawSurface

の

IDirectDraw *dwRefreshRate* *dwFlags*

IDirectDraw::SetDisplayMode **IDirectDraw::SetCooperativeLevel** のの

IDirectDraw2

IDirectDraw

IDirectDraw::RestoreDisplayMode

1 ”‘“ƒ ƒ,,,,,, DDSCL_EXCLUSIVEƒƒƒ,ƒƒƒ,, **IDirectDraw::SetCooperativeLevel**,Œ, ,

2 ƒƒƒƒƒƒ ƒ ƒ,• ,,,, **IDirectDraw::SetDisplayMode**,Œ, ,

3 ”‘“ƒ ƒ,‰ ,,,,,DDSCL_NORMALƒƒƒ,ƒƒƒ,,**IDirectDraw::SetCooperativeLevel** ,Œ, ,

IDirectDraw2

の

IDirectDraw2::RestoreDisplayMode

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

IDirectDraw2::WaitForVerticalBlank

```
HRESULT WaitForVerticalBlank(DWORD dwFlags, HANDLE hEvent);
```

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_UNSUPPORTED
DDERR_WASSTILLDRAWING

dwFlags
’‹ ŠŠ,‘, ,,,,,‘,,,Œ’,,

DDWAITVB_BLOCKBEGIN
’‹ ŠŠ,ŠŽŽ,,•,

DDWAITVB_BLOCKBEGINEVENT
’‹ ,Ž,,,,,ƒƒƒƒ ƒƒƒ ,,,Œ ƒƒ ƒ,,,,,,

DDWAITVB_BLOCKEND
’‹ ŠŠ, ,, •Ž,ŠŽ,,,,,•,,,

hEvent
’‹ ,Ž,,,,,,ƒƒƒ,,,ƒƒƒƒ ƒƒƒƒ
IDirectDraw2::GetVerticalBlankStatus **IDirectDraw2::GetScanLine**

IDirectDrawClipper

IDirectDrawClipper

IDirectDrawClipper

IDirectDrawClipper

ƒƒƒ”’	**Initialize**
ƒƒƒƒƒƒ ƒƒƒ	**GetClipList** **IsClipListChanged** **SetClipList** **SetHWnd**
ƒƒƒƒ	**GetHWnd**

IDirectDrawClipper COM
IUnknown **Iunknown**
の

AddRef
QueryInterface
Release

IDirectDrawClipper::GetClipList

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

DirectDrawClipper

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_GENERIC

  DDERR_INVALIDCLIPLIST

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_NOCLIPLIST

  DDERR_REGIONTOOSMALL

lpRect

ƒƒƒƒƒƒ ƒƒƒ,ƒƒƒƒƒƒ,,,,,Ž—,,‹Œ, ‘‘,,ƒƒƒƒ

lpClipList

ƒƒƒƒƒƒ ƒƒƒ,ƒƒ Œ‰,Š,**RGNDATA** ‘‘,,ƒƒƒƒ

lpdwSize

ƒƒƒƒƒƒ ƒƒƒ,ƒƒƒ, ’,,• ,,ƒƒƒƒ

RGNDATA の

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

RGNDATA の rdh の RGNDATAHEADER

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

のの Win32SDK の

IDirectDrawClipper::SetClipList

IDirectDrawClipper::GetHWnd

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

IDirectDrawClipper::SetHWnd DirectDrawClipper
¥

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS

lphWnd
,,,**IDirectDrawClipper::SetHWnd**ƒƒƒƒ,,,,
DirectDrawClipperƒƒƒƒƒƒ,Š˜,,,,,ƒƒƒƒƒ¥ƒƒƒƒ,ƒƒƒƒ

IDirectDrawClipper::SetHWnd

IDirectDrawClipper::Initialize

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

DirectDrawClipper **CoCreateInstance** OLE

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_ALREADYINITIALIZED
  DDERR_INVALIDPARAMS

lpDD
DirectDrawƒƒƒƒƒƒ, DirectDraw ‘‘,ƒƒƒƒ ,,ƒƒƒ ƒ,NULL,ƒƒƒ,,,, “—
,,DirectDrawClipperƒƒƒƒƒƒ, ,,, (**DirectDrawCreateClipper**Š , Ž,,,“,,
)

dwFlags
,,ƒƒƒ ƒ,Œ –Ž—,,, 0,ƒƒƒ,,,,,,

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

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

lpbChanged

Œ‰,Š”,,•   ,,ƒƒƒƒ ƒƒƒƒƒƒ ƒƒƒ,• ,,,   TRUE,ƒƒƒ,,

IDirectDrawClipper::SetClipList

```
HRESULT SetClipList(LPRGNDATA lpClipList, DWORD dwFlags);
```

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

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,
  DDERR_CLIPPERISUSINGHWND

  DDERR_INVALIDCLIPLIST

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_OUTOFMEMORY

lpClipList

—Œ,**RGNDATA** ‘‘,,ƒƒƒƒ ,,,,NULL ,,,DirectDrawClipper ,Š˜,,,,,ƒƒƒƒƒƒ ƒƒƒ,‘ , ,,,,,,’,NULL,,,, ƒƒƒƒƒƒ ƒƒƒ,   ,,,

dwFlags

,,ƒƒƒ ƒ,Œ –Ž—, 0,ƒƒƒ,,,,,,,,,,,

¥   DirectDrawClipper

IDirectDrawSurface2::BltFast

RGNDATA の

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

RGNDATA の **rdh** の **RGNDATAHEADER**

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

のの Win32 SDK の

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

IDirectDrawClipper::SetHWnd

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

¥

- Œ‚‚‚ DD_OK Ž”‚‚‚Ž‚ƒƒ ‚•‚
  DDERR_INVALIDCLIPLIST
  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_OUTOFMEMORY

dwFlags
‚‚ƒƒƒ ƒ‚Œ –Ž—‚ 0‚ƒƒƒ‚‚‚‚‚‚‚‚‚‚‚
hWnd
ƒƒƒƒƒƒ •‚Š‚ƒƒƒƒƒ¥ƒƒƒƒ

IDirectDrawClipper::GetHWnd

IDirectDrawPalette

DirectDrawPalette
IDirectDrawPalette
のの*DirectDrawPalette*

IDirectDrawPalette

ƒƒƒ”’	**Initialize**
ƒƒƒƒ‚”—	**GetCaps**
ƒƒƒƒ ƒƒƒƒ	**GetEntries** **SetEntries**

IDirectDrawPalette COM
IUnknown **Iunknown** 3 の

AddRef
QueryInterface
Release

IDirectDrawPalette::GetCaps

```
HRESULT GetCaps(LPDWORD lpdwCaps);
```

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

lpdwCaps

ƒƒƒƒ,”—,Ž“,,**DDCAPS** ‘‘,**dwPalCaps**ƒƒƒ,,,ƒƒƒ

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

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_NOTPALETTIZED

dwFlags

,,ƒƒƒ ƒ,Œ –Ž— 0,ƒƒƒ,,,,,,

dwBase

ƒƒƒƒ,ŠŽƒƒƒƒƒƒ

dwNumEntries

lpEntries,Ž’,,,ƒƒƒƒ,“‰,,ƒƒƒƒ ƒƒƒƒ, ,,,,,ƒƒƒƒ ƒƒƒƒ, , ,•,,, *dwStartingEntry*,’,,ƒƒƒ ƒ *dwCount*,ƒƒƒƒ,,,’,, ,,,,ƒƒƒ ƒ, **IDirectDrawPalette::SetEntries** ,,,,‰Ž,,,

lpEntries

ƒƒƒƒ ƒƒƒƒ,ƒƒƒƒ DDPCAPS_8BITENTRIESƒƒƒ,ƒƒƒ,,, ƒƒƒƒ ƒƒƒƒ,,ƒƒƒ– ,,, ,,ˆŠ,4ƒƒƒ–,,, Šƒƒ ƒƒ, Ž’,,,

IDirectDrawPalette::SetEntries

IDirectDrawPalette::Initialize

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

DirectDrawPalette

- **DDERR_ALREADYINITIALIZED**,•,

lpDD
DirectDrawƒƒƒƒƒƒ,ˆ–,,DirectDraw ‘‘,ƒƒƒƒ

dwFlags and *lpDDColorTable*
,,,,ƒƒƒ ƒ,Œ –Ž— 0,ƒƒƒ,,,,,,

COM のの 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,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_NOPALETTEATTACHED

  DDERR_NOTPALETTIZED

  DDERR_UNSUPPORTED

dwFlags
,,ƒƒƒ ƒ,Œ –Ž— 0,ƒƒƒ,,,,,,

dwStartingEntry
ƒƒƒ,,, ,,,ƒƒƒƒ

dwCount
• ,,,ƒƒƒƒ ƒƒƒƒ

lpEntries
ƒƒƒƒ ƒƒƒƒ,ƒƒƒƒ ,,ƒƒƒƒ ƒƒƒƒ, DDPCAPS_8BITENTRIES
ƒƒƒ,ƒƒƒ,,,, ƒƒƒƒ ƒƒƒƒ,,ƒƒƒ, ,,,,, ,4ƒƒƒ,,, Šƒƒ ƒƒ, Ž’,‹ ,,,,

IDirectDrawPalette::SetEntries
IDirectDrawSurface2::SetPalette

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

IDirectDrawSurface2

IDirectDrawSurface2
DirectDrawSurface ののの
の *DirectDrawSurface*

IDirectDrawSurface2

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

	UpdateOverlayZOrder
‘	**GetBltStatus** **GetFlipStatus**
f fff”—	**GetCaps**
f fff ffff	**GetClipper** **SetClipper**
f fff‹	**GetPixelFormat** **GetSurfaceDesc**
f fff ffff	**GetPalette** **SetPalette**

IDirectDrawSurface2 COM
IUnknown **IUnknown**
の

AddRef
QueryInterface
Release

IDirectDrawSurface2::AddAttachedSurface

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

の

- Œ,,, 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 の
( ) 3D Z ののの
(
)

ののの
Z の
のののの
のの **IDirectDrawSurface2::Flip** のの

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

IDirectDrawSurface2::AddOverlayDirtyRect

```
HRESULT AddOverlayDirtyRect(LPRECT lpRect);
```

IDirectDrawSurface2::UpdateOverlayDisplay

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT
  DDERR_INVALIDPARAMS
  DDERR_INVALIDSURFACETYPE
  DDERR_UNSUPPORTED

lpRect
,,•—,,, **RECT** ‘‘,,ƒƒƒƒ

IDirectDrawSurface2::UpdateOverlayDisplay

IDirectDrawSurface2::Blt

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

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  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
ƒƒƒƒ“‘,,“‘ ,ƒ ƒƒƒ ,‹Œ, ,,,‰‰,ˆ’,’‹,, **RECT** ‘‘,ƒƒƒƒ

lpDDSrcSurface
ƒƒƒƒ“‘‘ ,“‘Œ,,, DirectDraw ƒ ƒƒƒ,ƒƒƒƒ

lpSrcRect
ƒƒƒƒ“‘,,,“‘Œƒ ƒƒƒ ,‹Œ, ,,,‰‰, ,’‹,, **RECT** ‘‘,ƒƒƒƒ

dwFlags

DDBLT_ALPHADEST

ƒƒƒƒ ƒƒ ƒƒƒ,ƒƒƒƒ •, ,,ƒƒƒƒ“‘,ƒƒƒƒ ƒƒƒƒ,,,“‘ ƒ ƒƒƒ,ƒƒƒƒ,,,ƒƒƒ ƒ ƒƒƒƒ ƒ ƒƒƒ,,,,,,Ž—,,

DDBLT_ALPHADESTCONSTOVERRIDE

,,ƒƒƒƒ“‘,“‘ ƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘, **dwAlphaDestConst** ƒƒƒ,Ž—,,

DDBLT_ALPHADESTNEG

ƒƒƒƒ’,‘‰,,,“‘ ƒ ƒƒƒ,“–‰,,,,,•, 0,•“–

DDBLT_ALPHADESTSURFACEOVERRIDE

“‘ ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘,**lpDDSAlphaDest** ƒƒƒ,Ž—,,

DDBLT_ALPHAEDGEBLEND

ƒƒ ƒ ,‰Ž,‰‘,ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘,**dwAlphaEdgeBlend** ƒƒƒ,Ž—,,

DDBLT_ALPHASRC

ƒƒƒƒ ƒƒ ƒƒƒ,,ƒƒƒƒ •, ,,ƒƒƒƒ“‘,ƒƒƒƒ ƒƒƒƒ,,,“‘Œƒ ƒƒƒ,ƒƒƒƒ,,,ƒƒƒ ƒ ƒƒƒƒ ƒ ƒƒƒ,,,,,,Ž—,,

DDBLT_ALPHASRCCONSTOVERRIDE

“‘Œƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘, **dwAlphaSrcConst** ƒƒƒ,Ž—,,

DDBLT_ALPHASRCNEG

ƒƒƒƒ’,‘‰,,,“‘Œƒ ƒƒƒ,“–‰,,,,,•, 0,•“–

DDBLT_ALPHASRCSURFACEOVERRIDE

“‘Œ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘,**lpDDSAlphaSrc** ƒƒƒ,Ž—,,

DDBLT_ASYNC

Ž‚Ž‚ ‚FIFO ‚’‚‚‚ƒƒƒƒ“‘‚”“ŠŽ ‚‚ FIFO ƒ ƒƒƒƒ‚‹—ˆ‚—— ‰”‚‚‚‚ Œ‚ ‚‚Ž”‚‚

DDBLT_COLORFILL

“‘ ƒ ƒƒƒ ‚“‘ ‹Œ‚–‚‚ RGB ‚‚‚ **DDBLTFX** ‘‘‚**dwFillColor** ƒƒƒ‚Ž —‚‚

DDBLT_DDFX

‚‚ƒƒƒƒ“‘‚Ž—‚‚Œ‰‚Ž’‚‚‚‚ **DDBLTFX** ‘‘‚**dwDDFX** ƒƒƒ‚Ž—‚‚

DDBLT_DDROPS

Win32 API‚Š‚‚‚ƒƒƒ — ROPS ‚Ž’‚‚‚‚**DDBLTFX** ‘‘‚**dwDDROPS** ƒƒƒ‚Ž—‚‚

DDBLT_DEPTHFILL

“‘ Zƒƒƒƒ ƒ ƒƒƒ ‚“‘ ‹Œ‚–‚‚ “’‚‚**DDBLTFX** ‘‘‚**dwFillDepth** ƒƒƒ‚Ž—‚‚

DDBLT_KEYDEST

“‘ ƒ ƒƒƒ‚Š˜‚‚‚‚ƒƒ ƒ ‚Ž—‚‚

DDBLT_KEYDESTOVERRIDE

“‘ ƒ ƒƒƒ‚ƒƒ ƒ ‚‚‚ **DDBLTFX** ‘‘‚**dckDestColorkey** ƒƒƒ‚Ž—‚‚

DDBLT_KEYSRC

“‘Œƒ ƒƒƒ‚Š˜‚‚‚‚ƒƒ ƒ ‚Ž—‚‚

DDBLT_KEYSRCOVERRIDE

“‘Œƒ ƒƒƒ‚ƒƒ ƒ ‚‚‚ **DDBLTFX** ‘‘‚**dckSrcColorkey** ƒƒƒ‚Ž—‚‚

DDBLT_ROP

‚‚ƒƒƒƒ“‘‚‚‚‚ROP ‚ **DDBLTFX** ‘‘‚ **dwROP** ƒƒƒ‚Ž—‚‚ ‚‚‚‚ ROP ‚ Win32 API ‚’‹‚‚‚‚‚‚‚‚“‚‚‚‚

DDBLT_ROTATIONANGLE

ƒ ƒƒƒ‚‰“Š 1/100“’ˆ‚Ž’‚‚‚ ‚‚‚**DDBLTFX** ‘‘‚ **dwRotationAngle** ƒƒƒ‚Ž—‚‚

DDBLT_WAIT

ƒƒƒƒ“‘‚ƒƒ ‚ DDERR_WASSTILLDRAWING•’‚‚‚‚•‚‚‚‘‚ ‚‚‚ƒƒƒƒ“‘‚ ”‚‚‚‚‚‚‚‚‚‘‚ƒƒ ‚ ” Ž‘ ‚‚‚‚–‚

DDBLT_ZBUFFER

ƒ ƒƒƒ‚“‘Œ‚“‘ ‚‚ƒƒƒƒ‚‚‚ Zƒƒƒƒ ‚‚‚Zƒƒƒƒ‚ƒƒƒ ƒ‚‚‚‚**DDBLTFX** ‘‘‚ **dwZBufferOpCode** ƒƒƒ‚Ž—‚ Zƒƒƒƒ ƒƒƒƒ“‘‚Ž ‚‚

DDBLT_ZBUFFERDESTCONSTOVERRIDE

“‘ ‚‚‚‚ **DDBLTFX** ‘‘‚**dwZDestConst**‚‚‚ **dwZBufferOpCode** ƒƒƒ‚‚‚‚‚ Zƒƒƒƒ‚‚‚Zƒƒƒƒ ƒƒƒ ƒ‚‚‚Ž—‚ Zƒƒƒƒ ƒƒƒƒ“‘‚Ž ‚‚

DDBLT_ZBUFFERDESTOVERRIDE

“‘ ‚‚‚‚ **DDBLTFX** ‘‘‚ **lpDDSZBufferDest** ‚‚‚ **dwZBufferOpCode** ƒƒƒ‚‚‚‚‚ Zƒƒƒƒ‚‚‚Zƒƒƒƒ ƒƒƒ ƒ‚‚‚Ž—‚ Zƒƒƒƒ ƒƒƒƒ“‘‚Ž ‚‚

DDBLT_ZBUFFERSRCCONSTOVERRIDE

“‘Œ‚‚‚‚ **DDBLTFX** ‘‘‚ **dwZSrcConst** ‚‚‚**dwZBufferOpCode** ƒƒƒ‚‚‚‚Zƒƒƒƒ‚‚Zƒƒƒƒ ƒƒƒ ƒ‚‚Ž—‚ Zƒƒƒƒ ƒƒƒƒ“‘‚Ž ‚‚

DDBLT_ZBUFFERSRCOVERRIDE

“‘Œ‚‚‚ **DDBLTFX** ‘‘‚ **lpDDSZBufferSrc** ‚‚‚ **dwZBufferOpCode**
ƒƒƒ‚‚‚‚ Zƒƒƒƒ‚‚Zƒƒƒƒ ƒƒƒ ƒ‚‚Ž—‚ Zƒƒƒƒ ƒƒƒƒ“‘‚Ž ‚‚

lpDDBltFx

DDBLTFX ‘‘‚ƒƒƒƒ

IDirectDrawSurface2::Blt

DDBLT_WAIT の

IDirectDrawSurface2::BltBatch

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

のの **IDirectDrawSurface2::Blt**

- Œ‚‚ DD_OK Ž”‚‚Ž‚ƒƒ ‚•‚

  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

ƒƒƒƒ“‘‚‚‚ƒƒƒ ƒ‚’‹‚‚ ‚ **DDBLTBATCH** ‘‘‚ƒƒƒƒ *dwCount*

Ž ,,ƒƒƒƒ“‘ ,

dwFlags

,,ƒƒƒ ƒ,Œ –Ž— 0,ƒƒƒ,,,,,,

IDirectDrawSurface2::BltFast

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

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_EXCEPTION

DDERR_GENERIC

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_INVALIDRECT

DDERR_NOBLTHW

DDERR_SURFACEBUSY

DDERR_SURFACELOST

DDERR_UNSUPPORTED

dwX and *dwY*

“‘ ƒ ƒƒƒ ,ƒƒƒƒ“‘,,, •,,,, •

lpDDSrcSurface

ƒƒƒƒ“‘‘ ,“‘Œ DirectDraw ƒ ƒƒƒ,ƒƒƒƒ

lpSrcRect

ƒƒƒƒ“‘,,,“‘Œƒ ƒƒƒ ,‹Œ, ,,,‰‰, •,’‹,, **RECT** ‘‘,ƒƒƒƒ

dwTrans

“‘ƒƒƒ

DDBLTFAST_DESTCOLORKEY

“‘ ,ƒƒ ƒ ,Ž—,,“‰Œƒƒƒƒ“‘,Ž’,, .

DDBLTFAST_NOCOLORKEY

“‰,,,’ ,ƒƒ ƒƒƒƒ“‘,Ž’,,

DDBLTFAST_SRCCOLORKEY

“‘Œ,ƒƒ ƒ ,Ž—,,“‰Œƒƒƒƒ“‘,Ž’,,

DDBLTFAST_WAIT

ƒƒƒƒ“‘,ƒƒ ,
DDERR_WASSTILLDRAWING•’,,,,,•,,,‘, ,,,ƒƒƒƒ“‘, ”,,,,,,,,,‘,ƒƒ
,” Ž‘ ,,,,–,

のの

IDirectDrawSurface2::BltFast **IDirectDrawSurface2::Blt**

10

のの

IDirectDrawSurface2::BltFast

DDBLT_WAIT

IDirectDrawSurface2::DeleteAttachedSurface

```
HRESULT DeleteAttachedSurface(DWORD dwFlags,
    LPDIRECTDRAWSURFACE2 lpDDSAttachedSurface);
```

( )

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_CANNOTDETACHSURFACE

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_SURFACELOST

  DDERR_SURFACENOTATTACHED

dwFlags

,,ƒƒƒ ƒ,Œ –Ž—,,, 0,ƒƒƒ,,,,,,,,,,,,

lpDDSAttachedSurface

ƒƒƒƒ,,, DirectDraw ƒ ƒƒƒ,ƒƒƒƒ
,,ƒƒƒ ƒ,NULL, ƒƒƒƒ,,,,,,,,,,ƒ ƒƒƒ,ƒƒƒƒ,,,

IDirectDrawSurface2::AddAttachedSurface

DirectDraw

の

IDirectDrawSurface2::Flip

IDirectDrawSurface2::EnumAttachedSurfaces

```
HRESULT EnumAttachedSurfaces(LPVOID lpContext,
    LPDDENUMSURFACESCALLBACK lpEnumSurfacesCallback);
```

の

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_SURFACELOST

lpContext

ƒ ƒƒƒƒŠ ,“,,, ƒƒƒƒ ƒƒƒ,’‹,, ‘‘,,ƒƒƒƒ

lpEnumSurfacesCallback

ƒƒƒƒ,,,,,ƒ ƒƒƒ–,Œ, ,,, **EnumSurfacesCallback** Š ,ƒƒƒƒ

IDirectDrawSurface2::EnumOverlayZOrders

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

の

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

dwFlags

ˆ‰,ƒƒƒ,,,,,,,,,,’,Ž,

DDENUMOVERLAYZ_BACKTOFRONT

ƒƒƒ,,ƒƒƒƒ,Œ,,ƒ ƒ ƒƒ,—‹,,

DDENUMOVERLAYZ_FRONTTOBACK

ƒƒƒƒ,,ƒƒƒ,Œ,,ƒ ƒ ƒƒ,—‹,,

lpContext

Šƒ ƒ ƒƒ ƒ ƒƒƒ,ƒ ƒƒƒƒŠ ,“,,,ƒ ƒ’‹ƒƒƒƒƒƒ,,ƒƒƒƒ

lpfnCallback

,,ƒ ƒƒƒ ,ƒ ƒ ƒƒ,,,ƒ ƒƒƒ–,Œ, ,,,, **fnCallback** Š ,ƒƒƒƒ

IDirectDrawSurface2::Flip

```
HRESULT Flip(
    LPDIRECTDRAWSURFACE2 lpDDSurfaceTargetOverride,
    DWORD dwFlags);
```

DDSCAPS_BACKBUFFER

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_GENERIC

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_NOFLIPHW

DDERR_NOTFLIPPABLE

DDERR_SURFACEBUSY

DDERR_SURFACELOST

DDERR_UNSUPPORTED

DDERR_WASSTILLDRAWING

lpDDSurfaceTargetOverride
ƒƒƒƒƒƒ,,, DirectDraw ƒ ƒƒƒ,ƒƒƒƒ
,,ƒƒƒ ƒ,ƒƒƒƒƒ,NULL,,, ,,,,**IDirectDrawSurface2::Flip**,ƒƒƒƒ’,Œ,,ƒƒƒƒ,,
,,, ,ƒƒƒƒ,, ,,ƒƒƒ ƒ,ƒ ƒ ƒƒƒ,,,,,Ž—,,,

dwFlags
DDFLIP_WAIT
ˆ”, ƒƒƒƒƒƒ ƒ ƒƒƒƒ, ‘,“ ,,,ƒƒƒƒƒƒ, ”,,,,,
DDERR_WASSTILLDRAWING
ƒƒ ,,,,,•, ƒƒƒƒƒƒ,‹,,,, ,,, ,,ƒƒƒ,ƒƒƒ,,,
HAL,,DDERR_WASSTILLDRAWING
ƒƒ ,Ž,Ž,,,**IDirectDrawSurface2::Flip**,ƒƒƒƒƒƒ,,,,,‘,,
,, **IDirectDrawSurface2::Flip**,ƒƒƒƒƒƒ —, Œ,,,
DDERR_SURFACEBUSY,,,,•,ƒƒ ,•,,,,,–,,,

DDSCAPS_FLIP
DDSCAPS_FRONTBUFFER の
の
IDirectDrawSurface2::Flip

lpDDSurfaceTargetOverride

NULL

IDirectDrawSurface2::Flip

IDirectDrawSurface2::GetFlipStatus

IDirectDrawSurface2::GetAttachedSurface

```
HRESULT GetAttachedSurface(LPDDSCAPS lpDDSCaps,
    LPDIRECTDRAWSURFACE2 FAR * lplpDDAttachedSurface);
```

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_NOTFOUND

DDERR_SURFACELOST

lpDDSCaps
ƒ ƒƒƒ,ƒ ƒƒƒƒ”—,Ž, **DDSCAPS** ‘,ƒƒƒƒ

lplpDDAttachedSurface
EnumSurfacesCallback ƒ ƒƒƒƒŠ , *lpDDSurface*
ƒƒƒ ƒ‚Ž’‚‚Œ , DirectDraw ƒ ƒƒƒ‚ƒƒƒƒ‚‚ ‚‚*lpDDSCaps*
ƒƒƒ ƒ‚Ž’‚Œ ‚”—‚Ž‚‚ DirectDraw ƒ ƒƒƒ‚‚ƒƒƒƒ‚ƒƒƒƒ

DirectDrawSurface
Z ‘‘

‘

IDirectDrawSurface2::EnumAttachedSurfaces

IDirectDrawSurface2::GetBltStatus

`HRESULT GetBltStatus(DWORD dwFlags);`

- ƒƒƒƒ“‘ƒ ƒƒƒƒ‚‘ ‚‚‚‚ DD_OK
‚•‚ ƒƒƒƒ“‘ƒ ƒƒƒƒ‚ƒƒ ‚‚ DDERR_WASSTILLDRAWING
ƒƒƒƒ“‘ƒ ƒƒƒƒ‚‘ ‚‚‚‚‚ DDERR_NOBLTHW Ž”‚‚‚Ž‚ƒƒ ‚•‚

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_NOBLTHW

DDERR_SURFACEBUSY

DDERR_SURFACELOST

DDERR_UNSUPPORTED

DDERR_WASSTILLDRAWING

dwFlags
ˆ‰‚ƒƒƒ‚‚‚‚‚‚‚‚‚‚Ž‚

DDGBS_CANBLT

‚‚ƒ ƒƒƒ‚Š‚‚ƒƒƒƒ“‘‚‚‚‚‚Ž ‚‚‚‚‚‚‚‚– ‚‚ ƒƒƒƒ“‘‚Š—‚‚‚ DD_OK ‚•‚

DDGBS_ISBLTDONE

ƒƒƒƒ“‘‚ ‚‚‚‚‚‚‚‚– ‚‚ ‚‚ƒ ƒƒƒ‚ Œ‚ƒƒƒƒ“‘‚Š—‚‚‚‚‚‚DD_OK ‚•‚

IDirectDrawSurface2::GetCaps

`HRESULT GetCaps(LPDDSCAPS lpDDSCaps);`

‘

- Œ‚‚‚ DD_OK Ž”‚‚‚Ž‚ƒƒ ‚•‚

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

lpDDSCaps
ƒ ƒƒƒ‚ƒ ƒƒƒƒ”—‚“‚ **DDSCAPS** ‘‘‚ƒƒƒƒ

IDirectDrawSurface2::GetClipper

`HRESULT GetClipper(LPDIRECTDRAWCLIPPER FAR * lplpDDClipper);`

ƒ DirectDrawClipper

- Œ‚‚‚ DD_OK Ž”‚‚‚Ž‚ƒƒ ‚•‚

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_NOCLIPPERATTACHED

lplpDDClipper
ƒ ƒƒƒ‚Š˜‚‚‚‚‚ DirectDrawClipper ƒƒƒƒƒƒ‚‚ƒƒƒƒ‚ƒƒƒƒ

IDirectDrawSurface2::SetClipper

IDirectDrawSurface2::GetColorKey

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

DirectDrawSurface ƒ

- Œ‚‚‚ DD_OK Ž”‚‚‚Ž‚ƒƒ ‚•‚

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_NOCOLORKEY

DDERR_NOCOLORKEYHW

DDERR_SURFACELOST

DDERR_UNSUPPORTED

dwFlags
‚‚ƒƒ ƒ ‚—‹‚‚‚‚Ž’‚‚

DDCKEY_DESTBLT

‘‘‚ƒƒ ƒ ‚‚‚‚ƒƒƒƒ“‘‚‚‚‚“‘ ƒƒ ƒ ‚‚‚Ž‚‚‚ƒƒ ƒƒ ƒ‚Ž’‚‚

DDCKEY_DESTOVERLAY

‘‘‚ƒƒ ƒ ‚‚‚‚ƒ ƒ ƒƒ —
‚‚‚‚“‘ ƒƒ ƒ ‚‚‚Ž‚‚‚ƒƒ ƒƒ ƒ‚Ž’‚‚

DDCKEY_SRCBLT

‘‘‚ƒƒ ƒ ‚‚‚‚ƒƒƒƒ“‘‚‚‚‚“‘Œƒƒ ƒ ‚‚‚Ž‚‚‚ƒƒ ƒƒ ƒ‚Ž’‚‚

DDCKEY_SRCOVERLAY

‘‘‚ƒƒ ƒ ‚‚‚‚ƒ ƒ ƒƒ —‚‚‚‚“‘Œƒƒ ƒ ‚‚‚Ž‚‚‚ƒƒ ƒƒ ƒ‚Ž’‚‚

lpDDColorKey
DirectDrawSurface ƒƒƒƒƒƒ,Ž’,,,ƒƒ ƒ ,Œ ,’,Š”,,**DDCOLORKEY**
‘‘,ƒƒƒƒ

IDirectDrawSurface2::SetColorKey

IDirectDrawSurface2::GetDC

```
HRESULT GetDC(HDC FAR * lphDC);
```

のの GDI

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_DCALREADYCREATED

DDERR_GENERIC

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_INVALIDSURFACETYPE

DDERR_SURFACELOST

DDERR_UNSUPPORTED

DDERR_WASSTILLDRAWING

lphDC
ƒƒƒƒ ƒƒƒƒƒƒ,•,,,ƒƒƒƒ,,ƒƒƒƒ

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

IDirectDrawSurface2::Lock

IDirectDrawSurface2::GetDDInterface

```
HRESULT GetDDInterface(LPVOID FAR *lplpDD);
```

DirectDraw

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

lplpDD
DirectDraw ƒƒƒƒ,•,,,ƒƒƒƒ,ƒƒƒƒ Œ, ,,Ž”,, ,,’,–Œ,,,
IDirectDraw2 **IDirectDraw**

IDirectDrawSurface2::GetFlipStatus

```
HRESULT GetFlipStatus(DWORD dwFlags);
```

- Œ,,, DD_OK ƒ ƒƒƒ,ƒƒƒƒƒƒ —, ,,,,,,,
  DDERR_WASSTILLDRAWING ,,‘,——,Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_INVALIDSURFACETYPE

  DDERR_SURFACEBUSY

  DDERR_SURFACELOST

  DDERR_UNSUPPORTED

dwFlags
ˆ‰,ƒƒƒ,,,,,,,Ž’,,

DDGFS_CANFLIP

,,ƒ ƒƒƒ,,,,,ƒƒƒƒƒƒ,,,,,,,,–, ,, ƒƒƒƒƒƒ,Š—,,, DD_OK,•,

DDGFS_ISFLIPDONE

ƒƒƒƒƒƒ, ,,,,,,,–, ,, ,,ƒ ƒƒƒ, Œ,ƒƒƒƒƒƒ,Š—,,,, ,,DD_OK ,•,

IDirectDrawSurface2::Flip

IDirectDrawSurface2::GetOverlayPosition

```
HRESULT GetOverlayPosition(LPLONG lplX, LPLONG lplY);
```

の DDSCAPS_OVERLAY のの

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_GENERIC

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_INVALIDPOSITION

  DDERR_NOOVERLAYDEST

  DDERR_NOTAOVERLAYSURFACE

  DDERR_OVERLAYNOTVISIBLE

  DDERR_SURFACELOST

lplX and *lplY*
xƒƒƒƒƒƒ •,,,yƒƒƒƒƒƒ •,ƒƒƒƒ

IDirectDrawSurface2::SetOverlayPosition
IDirectDrawSurface2::UpdateOverlay

IDirectDrawSurface2::GetPalette

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

DirectDrawPalette

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_GENERIC

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_NOEXCLUSIVEMODE

DDERR_NOPALETTEATTACHED

DDERR_SURFACELOST

DDERR_UNSUPPORTED

lplpDDPalette

Ž“,,ƒ ƒƒƒ,Š˜,,,,, DirectDrawPalette ‘‘,ƒƒƒƒ,,ƒƒƒƒ ,,ƒ ƒƒƒ,Š˜,,,,, DirectDrawPalette ‘‘,,, NULL ,ƒƒƒ,,, ,,ƒƒƒ ƒ, ƒ ƒƒƒ,ƒƒƒƒƒ ƒ ƒƒƒ, ,,,,ƒƒƒƒƒ ƒ ƒƒƒ,ƒƒƒ ƒƒƒƒ, ,,ƒƒ ƒƒƒ ƒ ƒƒƒ,8 bppƒ ƒ,,, ƒƒƒƒ ƒƒƒƒ,,ƒƒƒƒ,Ž,

IDirectDrawSurface2::SetPalette

IDirectDrawSurface2::GetPixelFormat

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

の

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_INVALIDSURFACETYPE

lpDDPixelFormat

Œ ƒƒƒƒ, •,ƒ ƒƒƒ,ƒƒ ƒƒ ƒ ƒƒ ƒƒƒ, ,‹ ,Ž“,, **DDPIXELFORMAT** ‘‘,,ƒƒƒƒ

IDirectDrawSurface2::GetSurfaceDesc

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

のの **DDSURFACEDESC**

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

lpDDSurfaceDesc
ƒ ƒƒƒ, ‘,Ž“,,**DDSURFACEDESC** ‘‘,,ƒƒƒƒ
DDSURFACEDESC

IDirectDrawSurface2::Initialize

```
HRESULT Initialize(LPDIRECTDRAW lpDD,
    LPDDSURFACEDESC lpDDSurfaceDesc);
```

DirectDrawSurface

- **DDERR_ALREADYINITIALIZED**,•,

lpDD
DirectDraw ƒƒƒƒƒƒ,•, DirectDraw ‘‘,ƒƒƒƒ
lpDDSurfaceDesc
ƒ ƒƒƒ,,,,, •, ’,,, **DDSURFACEDESC** ‘‘,,ƒƒƒƒ
の COM
DirectDrawSurface
DDERR_ALREADYINITIALIZED

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

IDirectDrawSurface2::IsLost

```
HRESULT IsLost();
```

DirectDrawSurface

- Ž,,,,,,,, DD_OK ,,,,,,,Ž,ƒƒ ,•,
  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

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_OUTOFMEMORY

  DDERR_SURFACEBUSY

  DDERR_SURFACELOST

  DDERR_WASSTILLDRAWING

lpDestRect
ƒƒƒ,,ƒ ƒƒƒ—ˆ,Ž, **RECT** ‘‘,ƒƒƒƒ

lpDDSurfaceDesc
ƒ ƒƒƒ,,,,, •,Ž’,, **DDSURFACEDESC** ‘‘,,ƒƒƒƒ

dwFlags

DDLOCK_EVENT

IDirectDrawSurface2::Lock,—‹,,,ƒ ƒƒƒ ƒƒƒ,ƒƒƒƒ,•,,,,‰”,,,,ƒƒƒƒ,‹ “,,ƒƒƒ ƒƒƒƒ ƒƒƒƒ,**IDirectDrawSurface2::Lock** ,“,,,,, ƒƒƒ,,, ,,ƒƒƒ,ƒƒƒ,ƒ ƒƒƒ ,• ,,,, ƒƒƒƒ,FIFO ,ˆ,‹,,,,

DDLOCK_READONLY

ƒƒƒ,,ƒ ƒƒƒ,“, , —,,,,,,Ž,ƒƒƒ

DDLOCK_SURFACEMEMORYPTR

Ž’,,‹Œ, “,,—Œ,ƒƒƒ ƒƒƒƒ,•,,,,,,,,,,,,•,ƒƒƒ ‹Œ,Ž’,,,, ˆ” ,ƒ ƒƒƒ,,ƒƒƒƒ,•,,, ƒƒƒƒƒ,,,,ƒƒƒ,Ž’,,

DDLOCK_WAIT

’ ƒƒƒƒ“‘ —,ƒƒ ‘,,,ƒƒƒ,“,,,, ƒƒƒƒ,’,, —, DDERR_WASSTILLDRAWING ƒƒ ,•, ,,,,,ƒƒƒ,ƒƒƒ,, ƒƒƒ,“,,,, DDERR_SURFACEBUSY,, •,ƒƒ ,” ,,,, **IDirectDrawSurface2::Lock**, Ž ,,

DDLOCK_WRITEONLY

ƒƒƒ,,ƒ ƒƒƒ, ,,, —,,,,,,Ž,

hEvent
ƒ ƒƒƒ,ƒƒƒ, ”,,,, Š,ƒƒƒ,,,ƒƒƒƒ ƒƒƒƒ,,ƒƒƒƒ

IDirectDrawSurface2::Unlock

IDirectDrawSurface2::Unlock

の

DDERR_SURFACEBUSY DDERR_LOCKEDSURFACES

IDirectDrawSurface2::Lock
DDLOCK_WAIT

のの DirectDraw
IDirectDrawSurface2::Lock **IDirectDrawSurface2::Unlock**
Win16 Win16 GDI USER

Windows **IDirectDrawSurface2::Lock/IDirectDrawSurface2::Unlock**
IDirectDrawSurface2::GetDC/IDirectDrawSurface2::ReleaseDC の
Windows GUI の

IDirectDrawSurface2::Unlock **IDirectDrawSurface2::GetDC**
IDirectDrawSurface2::ReleaseDC

IDirectDrawSurface2::PageLock

```
HRESULT PageLock(DWORD dwFlags);
```

(DMA) のの

- Œ,,, DD_OK Ž",,,Ž,ƒƒ ,•,

DDERR_CANTPAGELOCK
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_SURFACELOST

dwFlags
,,ƒƒƒ ƒ,Œ –Ž— 0,ƒƒƒ,,,,,,

のの**IDirectDrawSurface2::PageLock**
IDirectDrawSurface2::PageUnlock
0

ののの
DD_OK

IDirectDraw2 **IDirectDraw**

IDirectDrawSurface2::PageUnlock

IDirectDrawSurface2::PageUnlock

```
HRESULT PageUnlock(DWORD dwFlags);
```

の

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_CANTPAGEUNLOCK

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_NOTPAGELOCKED

  DDERR_SURFACELOST

dwFlags
,,ƒƒƒ ƒ,Œ –Ž— 0,ƒƒƒ,,,,,,

の**IDirectDrawSurface2::PageLock**
IDirectDrawSurface2::PageUnlock 0
の

ののの の
DD_OK

IDirectDraw2 **IDirectDraw**

IDirectDrawSurface2::PageLock

IDirectDrawSurface2::ReleaseDC

```
HRESULT ReleaseDC(HDC hDC);
```

IDirectDrawSurface2::GetDC

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_GENERIC

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_SURFACELOST

  DDERR_UNSUPPORTED

hDC
IDirectDrawSurface2::GetDCƒƒƒƒ,Ž—,,Ž“,,ƒƒƒƒ ƒƒƒƒƒƒ,ƒƒƒƒ
IDirectDrawSurface2::GetDC
の

IDirectDrawSurface2::GetDC

IDirectDrawSurface2::Restore

```
HRESULT Restore();
```

DirectDrawSurface

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_GENERIC

  DDERR_IMPLICITLYCREATED

  DDERR_INCOMPATIBLEPRIMARY

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_NOEXCLUSIVEMODE

  DDERR_OUTOFMEMORY

  DDERR_UNSUPPORTED

  DDERR_WRONGMODE

のので
DirectDrawSurface のの
DDERR_SURFACELOST
IDirectDrawSurface2::Restore
DirectDrawSurface

DirectDrawSurface

IDirectDrawSurface2::Restore
IDirectDrawSurface2::AddAttachedSurface

IDirectDrawSurface2::IsLost
IDirectDrawSurface2::AddAttachedSurface

IDirectDrawSurface2::SetClipper

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

DirectDrawClipper DirectDrawSurface

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_INVALIDSURFACETYPE

  DDERR_NOCLIPPERATTACHED

lpDDClipper
DirectDrawSurface ƒƒƒƒƒƒ,ƒƒƒƒ,, DirectDrawClipper ƒƒƒƒƒƒ,•,,,, DirectDrawClipper ‘‘,ƒƒƒƒ ,,,,ƒƒƒ ƒ,ƒƒ,,, Œ , DirectDrawClipper ƒƒƒƒƒƒ,ƒƒƒƒ,,,,,ˆ–,,

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

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_GENERIC

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_INVALIDSURFACETYPE

  DDERR_NOOVERLAYHW

  DDERR_NOTAOVERLAYSURFACE

  DDERR_SURFACELOST

  DDERR_UNSUPPORTED

  DDERR_WASSTILLDRAWING

dwFlags
,,,,ƒƒ ƒ ,—‹,,,,Ž’,,

DDCKEY_COLORSPACE

‘‘,ƒƒ ƒƒ ƒ,Š,,,, ƒƒƒ,, ‘‘,’ˆ,ƒƒ ƒ ,Š,,,,,,,ƒƒƒ,,,,,,,

DDCKEY_DESTBLT

ƒƒƒƒ“‘,“‘ ƒƒ ƒ ,,,Ž—
,,,ƒƒ ƒ ,,,,“‘ ƒƒ ƒƒ ƒ, ‘‘,Ž’,,,, ƒƒƒ,,

DDCKEY_DESTOVERLAY

ƒ ƒ ƒƒ —,“‘ ƒƒ ƒ ,,,Ž—
,,,ƒƒ ƒ ,,,,“‘ ƒƒ ƒƒ ƒ, ‘‘,Ž’,,,, ƒƒƒ,,

DDCKEY_SRCBLT

ƒƒƒƒ“‘,“‘Œƒƒ ƒ ,,,Ž—
,,,ƒƒ ƒ ,,,,“‘ ƒƒ ƒƒ ƒ, ‘‘,Ž’,,,, ƒƒƒ,,

DDCKEY_SRCOVERLAY

ƒ ƒ ƒƒ —,“‘Œƒƒ ƒ ,,,Ž—
,,,ƒƒ ƒ ,,,,“‘ ƒƒ ƒƒ ƒ, ‘‘,Ž’,,,, ƒƒƒ,,

lpDDColorKey

DirectDrawSurface ƒƒƒƒƒƒ, ,,ƒƒ ƒ ’,Š,,,, **DDCOLORKEY** ‘‘,ƒƒƒƒ

IDirectDrawSurface2::GetColorKey

IDirectDrawSurface2::SetOverlayPosition

```
HRESULT SetOverlayPosition(LONG lX, LONG lY);
```

の

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_GENERIC
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_SURFACELOST
DDERR_UNSUPPORTED

lX and *lY*

,,,ƒƒƒƒƒƒ •,,,,ƒƒƒƒƒƒ •

IDirectDrawSurface2::GetOverlayPosition
IDirectDrawSurface2::UpdateOverlay

IDirectDrawSurface2::SetPalette

```
HRESULT SetPalette(LPDIRECTDRAWPALETTE lpDDPalette);
```

DirectDrawPalette のの

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_GENERIC
DDERR_INVALIDOBJECT
DDERR_INVALIDPARAMS
DDERR_INVALIDSURFACETYPE
DDERR_NOEXCLUSIVEMODE
DDERR_NOPALETTEATTACHED
DDERR_NOPALETTEHW
DDERR_NOT8BITCOLOR
DDERR_SURFACELOST
DDERR_UNSUPPORTED

lpDDPalette

DirectDrawPalette ‘‘,ƒƒƒƒ

ののの

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

IDirectDrawSurface2::Unlock

```
HRESULT Unlock(LPVOID lpSurfaceData);
```

DirectDraw

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_GENERIC

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_INVALIDRECT

DDERR_NOTLOCKED

DDERR_SURFACELOST

lpSurfaceData

IDirectDrawSurface2::Lockƒƒƒƒ,,,,•,,,ƒƒƒƒ,ƒƒƒƒ “,ƒ ƒƒƒ,ˆ,,“‘ ‹Œ,,,,• , **IDirectDrawSurface2::Lock**Œ, ,,‰”,,, ,,ƒƒƒƒ, **IDirectDrawSurface2::Lock**,,, **IDirectDrawSurface2::Unlock**Œ, ,,,Œ,•,,

IDirectDrawSurface2::Lock

IDirectDrawSurface2::UpdateOverlay

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

の の

DDSCAPS_OVERLAY

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

DDERR_GENERIC

DDERR_HEIGHTALIGN

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_INVALIDRECT

DDERR_INVALIDSURFACETYPE

DDERR_NOSTRETCHHW

DDERR_NOTAOVERLAYSURFACE

DDERR_SURFACELOST

DDERR_UNSUPPORTED

DDERR_XALIGN

lpSrcRect

“‘Œƒ ƒƒƒ‚ƒ ƒ ƒƒ‚Ž‚ ‚• ‚•‚‚‚—ˆ‚ ‚‚’‹‚‚ **RECT** ‘‘‚ƒƒƒƒ

lpDDDestSurface

ƒ ƒ ƒƒ‚‚‚‚‚ DirectDraw ƒ ƒƒƒ‚ƒƒƒƒ

lpDestRect

ƒ ƒ ƒƒ‚ ‚‚‚“‘ ƒ ƒƒƒ‚ ‚ ‚• ‚•‚‚‚—ˆ‚ ‚‚’‹‚‚ **RECT** ‘‘‚ƒƒƒƒ

dwFlags

DDOVER_ADDDIRTYRECT

ƒƒƒƒ ƒƒƒ‚ƒ ƒ ƒƒ‚‚‚ƒ ƒƒƒ‚ ‚‚‚‚‚‚‚‚—ˆ‚‰‚‚

DDOVER_ALPHADEST

ƒƒƒƒ ƒƒ ƒƒƒ‚‚ƒƒƒƒ •‚‚‚‚ƒƒƒƒ ƒƒƒƒ ƒ ƒƒƒ‚ ‚‚ƒ ƒ ƒƒ‚“‘ ƒƒƒƒ ƒƒƒƒ‚‚‚ƒ ƒƒƒ‚ƒƒƒƒ‚‚‚‚Ž—‚‚

DDOVER_ALPHADESTCONSTOVERRIDE

‚‚ƒ ƒ ƒƒ‚“‘ ƒƒƒƒ ƒƒƒƒ‚‚‚**DDOVERLAYFX** ‘‘‚ **dwAlphaDestConst** ƒƒƒ‚Ž—‚‚

DDOVER_ALPHADESTNEG

ƒƒƒƒ’‚‘‰‚‚‚“‘ ƒ ƒƒƒ‚“–‰‚‚‚‚‚•‚ 0‚•“–

DDOVER_ALPHADESTSURFACEOVERRIDE

‚‚ƒ ƒ ƒƒ‚“‘ ƒƒƒƒ ƒƒƒƒ‚‚‚**DDOVERLAYFX** ‘‘‚ **lpDDSAlphaDest** ƒƒƒ‚Ž—‚‚

DDOVER_ALPHAEDGEBLEND

ƒƒ ƒ ‚‹‚ ‚‚‰‘‚ƒƒƒ‚ƒƒƒƒ ƒƒƒƒ‚‚‚**DDOVERLAYFX** ‘‘‚ **dwAlphaEdgeBlend** ƒƒƒ‚Ž—‚‚

DDOVER_ALPHASRC

ƒƒƒƒ ƒƒ ƒƒƒ‚‚ƒƒƒƒ •‚‚‚‚ƒƒƒƒ ƒƒƒƒ ƒ ƒƒƒ‚ ‚‚ƒ ƒ ƒƒ‚“‘Œƒƒƒƒ ƒƒƒƒ‚‚‚ƒ ƒƒƒ‚ƒƒƒƒ‚‚‚‚Ž—‚‚

DDOVER_ALPHASRCCONSTOVERRIDE

‚‚ƒ ƒ ƒƒ‚“‘Œƒƒƒƒ ƒƒƒƒ‚‚‚ **DDOVERLAYFX** ‘‘‚**dwAlphaSrcConst** ƒƒƒ‚Ž—‚‚

DDOVER_ALPHASRCNEG

ƒƒƒƒ’‚‘‰‚‚‚“‘Œƒ ƒƒƒ‚“–‰‚‚‚‚‚•‚ 0‚•“–

DDOVER_ALPHASRCSURFACEOVERRIDE

‚‚ƒ ƒ ƒƒ‚ƒƒƒƒ ƒƒƒƒŒ‚‚‚ **DDOVERLAYFX** ‘‘‚**lpDDSAlphaSrc** ƒƒƒ‚Ž—‚‚

DDOVER_DDFX

“Ž‚ƒ ƒ ƒƒŒ‰‚’‹‚‚‚‚‚ ƒ ƒ ƒƒFXƒƒƒ‚Ž—‚‚

DDOVER_HIDE

‚‚ƒ ƒ ƒƒ‚ƒƒ‚‚‚

DDOVER_KEYDEST

“‘ ƒ ƒƒƒ‚Œ‚•‚‚‚‚ƒƒ   ƒ ‚Ž—‚‚

DDOVER_KEYDESTOVERRIDE

“‘ ƒ ƒƒƒ‚ƒƒ   ƒ ‚‚‚ **DDOVERLAYFX** ‘‘‚**dckDestColorkey** ƒƒƒ‚Ž—‚‚

DDOVER_KEYSRC

“‘Œƒ ƒƒƒ‚Œ‚•‚‚‚‚ƒƒ   ƒ ‚Ž—‚‚

DDOVER_KEYSRCOVERRIDE

“‘Œƒ ƒƒƒ‚ƒƒ   ƒ ‚‚‚ **DDOVERLAYFX** ‘‘‚**dckSrcColorkey** ƒƒƒ‚Ž—‚‚

DDOVER_SHOW

‚‚ƒ ƒ ƒƒ‚ƒƒ‚‚‚

DDOVER_ZORDER

‚‚ƒ ƒ ƒƒ‚Zƒ ƒ ‚‚‚ **DDOVERLAYFX** ‘‘‚**dwZOrderFlags** ƒƒƒ‚Ž—‚‚ **lpDDSRelative** ƒƒƒ‚ **dwZOrderFlags** ƒƒƒ‚DDOVERZ_INSERTINBACKOF ‚‚‚‚DDOVERZ_INSERTINFRONTOF ‚ƒƒƒ‚‚‚‚‚Ž—‚‚

lpDDOverlayFx

DDOVERLAYFX ‘‘‚Ž ‚‚‚‚

IDirectDrawSurface2::UpdateOverlayDisplay

```
HRESULT UpdateOverlayDisplay(DWORD dwFlags);
```

ののの
の

- Œ‚‚‚ DD_OK Ž”‚‚‚Ž‚ƒƒ ‚•‚

DDERR_INVALIDOBJECT

DDERR_INVALIDPARAMS

DDERR_INVALIDSURFACETYPE

DDERR_UNSUPPORTED

dwFlags

—‚‚‚   ƒƒƒ ˆ‰‚ƒƒƒ‚‚‚‚‚‚’‚Ž‚

DDOVER_REFRESHDIRTYRECTS

‚‚“‘ ‚‚‚‚‘‚‚‚—ˆ‚‚   ‚‚‚‹Œ‚ƒƒƒ‚Ž— ‚‚ ƒƒƒƒƒƒ‚   ‚‚ ‚‚ ‚‚‹Œƒƒƒ‚ƒƒƒ‚‚

DDOVER_REFRESHALL

‚‚‚‹Œ‚ƒƒƒ‚–Ž‚‚ ƒ ƒ ƒƒ ƒƒƒƒƒƒ‚Š‘‚   ‚‚ ‚‚ ‚‚‹Œƒƒƒ‚ƒƒƒ‚‚

IDirectDrawSurface2::AddOverlayDirtyRect

IDirectDrawSurface2::UpdateOverlayZOrder

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

Z

- Œ,,, DD_OK Ž”,,,Ž,ƒƒ ,•,

  DDERR_INVALIDOBJECT

  DDERR_INVALIDPARAMS

  DDERR_NOTAOVERLAYSURFACE

dwFlags

ˆ‰,ƒƒƒ,,,,,,,’,Ž,

DDOVERZ_INSERTINBACKOF

,,ƒ ƒ ƒƒ,ƒ ƒ ƒƒ ƒƒ ƒ,Ž ƒ ƒ ƒƒ,Œ,,‘“,,

DDOVERZ_INSERTINFRONTOF

,,ƒ ƒ ƒƒ,ƒ ƒ ƒƒ ƒƒ ƒ,Ž ƒ ƒ ƒƒ,‘,‘“,,

DDOVERZ_MOVEBACKWARD

,,ƒ ƒ ƒƒ,ƒ ƒ ƒƒ ƒƒ ƒ,,,,Œ•,ˆ“,,

DDOVERZ_MOVEFORWARD

,,ƒ ƒ ƒƒ,ƒ ƒ ƒƒ ƒƒ ƒ,,,,‘•,ˆ“,,

DDOVERZ_SENDTOBACK

,,ƒ ƒ ƒƒ,ƒ ƒ ƒƒ ƒƒ ƒ,Œ•,ˆ“,,

DDOVERZ_SENDTOFRONT

,,ƒ ƒ ƒƒ,ƒ ƒ ƒƒ ƒƒ ƒ,‘•,ˆ“,,

lpDDSReference

ƒ ƒ ƒƒ ƒƒ ƒ,,,,‘“,ˆ’,,,Ž—,,,DirectDraw ƒ ƒƒƒ,ƒƒƒƒ ,,ƒƒƒ ƒ, DDOVERZ_INSERTINBACKOF,,, DDOVERZ_INSERTINFRONTOF ,,,,,,•—,,,,

IDirectDrawSurface2::EnumOverlayZOrders

□“

DDBLTBATCH

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

IDirectDrawSurface2::BltBatch

lprDest

ƒƒƒƒ“‘ ,’‹,, **RECT** ‘‘,ƒƒƒƒ

lpDDSSrc

ƒƒƒƒ“‘Œ,’‹,, DirectDrawSurfaceƒƒƒƒƒƒ,ƒƒƒƒ

lprSrc

ƒƒƒƒ“‘Œ,‹Œ,’‹,, **RECT** ‘‘,ƒƒƒƒ

dwFlags

ƒƒƒƒƒ Œƒƒƒ

DDBLT_ALPHADEST

,,ƒƒƒƒ“‘,,,,ƒƒƒƒ“‘ ƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, ƒƒƒƒ ƒƒ ƒƒƒ,,ƒƒƒƒ •,,,,ƒ ƒƒƒ ƒƒƒƒ ƒ ƒƒƒ,,,,,,Ž—,,

DDBLT_ALPHADESTCONSTOVERRIDE

,,ƒƒƒƒ“‘,,,,ƒƒƒƒ“‘ ƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘, **dwAlphaDestConst** ƒƒƒ,Ž—,,

DDBLT_ALPHADESTNEG

ƒƒƒƒ’,‘‰,,,“‘ ƒ ƒƒƒ,“–‰,,,,,•, 0,•“–

DDBLT_ALPHADESTSURFACEOVERRIDE

,,ƒƒƒƒ“‘,,,,ƒƒƒƒ“‘ ƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘, **lpDDSAlphaDest** ƒƒƒ,Ž—,,

DDBLT_ALPHAEDGEBLEND

ƒƒ ƒ ,‹, ,,‰‘,ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,,**DDBLTFX** ‘‘, **dwAlphaEdgeBlend** ƒƒƒ,Ž—,,

DDBLT_ALPHASRC

,,ƒƒƒƒ“‘,,,,ƒƒƒƒ“‘ ƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, ƒƒƒƒ ƒƒ ƒƒƒ,,ƒƒƒƒ •,,,,ƒ ƒƒƒ ƒƒƒƒ ƒ ƒƒƒ,,,,,,Ž—,,

DDBLT_ALPHASRCCONSTOVERRIDE

,,ƒƒƒƒ“‘,,,,ƒƒƒƒ“‘Œƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘, **dwAlphaSrcConst** ƒƒƒ,Ž—,,

DDBLT_ALPHASRCNEG

ƒƒƒƒ’,‘‰,,,“‘Œƒ ƒƒƒ,“–‰,,,,,•, 0,•“–

DDBLT_ALPHASRCSURFACEOVERRIDE

,,ƒƒƒƒ“‘,,,,ƒƒƒƒ“‘Œƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,,, **DDBLTFX** ‘‘, **lpDDSAlphaSrc** ƒƒƒ,Ž—,,

DDBLT_ASYNC

FIFO ƒ ƒƒƒƒ,’,,Ž ,, , ,,ƒƒƒƒ“‘,”“Š,Ž ,, ,, FIFO ƒ ƒƒƒƒ,‹—ˆ,,,,, Œ, ,,Ž”,,

DDBLT_COLORFILL

ƒƒƒƒ“‘ ƒ ƒƒƒ ,ƒƒƒƒ“‘ ‹Œ,–,, RGB ,,, **DDBLTFX** ‘‘, **dwFillColor** ƒƒƒ,Ž—,,

DDBLT_DDFX

,,ƒƒƒƒ“‘,Ž—,,,Œ‰,Ž’,, **DDBLTFX** ‘‘, **dwDDFX** ƒƒƒ,Ž—,,

DDBLT_DDROPS

Win32 API ,ˆ•,,,ƒƒƒ — ROPs ,Ž’,,,,,**DDBLTFX** ‘‘,

dwDDROPS ƒƒƒ,Ž—,,

DDBLT_KEYDEST

ƒƒƒƒ“‘ ƒ ƒƒƒ,Œ,•,,,,ƒƒ ƒ ,Ž—,,

DDBLT_KEYDESTOVERRIDE

ƒƒƒƒ“‘ ƒ ƒƒƒ,ƒƒ ƒ ,,, **DDBLTFX** ‘‘,**dckDestColorkey** ƒƒƒ,Ž—,,

DDBLT_KEYSRC

ƒƒƒƒ“‘Œƒ ƒƒƒ,Œ,•,,,,ƒƒ ƒ ,Ž—,,

DDBLT_KEYSRCOVERRIDE

•Œƒ ƒƒƒ,ƒƒ ƒ ,,, **DDBLTFX** ‘‘,**dckSrcColorkey** ƒƒƒ,Ž—,,

DDBLT_ROP

,,ƒƒƒƒ“‘, ROP ,,,, **DDBLTFX** ‘‘, **dwROP** ƒƒƒ,Ž—,, ROP , Win32 API ,’‹,,,,,,“,,,,

DDBLT_ROTATIONANGLE

ƒ ƒƒƒ,‰“Š 1/100“’ˆ,Ž’,,, ,,, **DDBLTFX** ‘‘, **dwRotationAngle** ƒƒƒ,Ž—,,

DDBLT_ZBUFFER

ƒƒƒƒ“‘Œ,,,ƒƒƒƒ“‘ ƒ ƒƒƒ,ƒƒƒƒ,,,Zƒƒƒƒ,Ž—,, Zƒƒƒƒ ƒƒƒƒ“‘, , **DDBLTFX** ‘‘, **dwZBufferOpCode** ƒƒƒ, Zƒƒƒƒ ƒƒƒ ƒ,,,

DDBLT_ZBUFFERDESTCONSTOVERRIDE

DDBLTFX ‘‘,**dwZDestConst**,“‘ Zƒƒƒƒ,Ž— ,, Zƒƒƒƒ ƒƒƒƒ“‘,Ž ,, “ ‘‘,**dwZBufferOpCode**ƒƒƒ,“‘ Zƒƒƒƒ ƒƒƒ ƒ,,,

DDBLT_ZBUFFERDESTOVERRIDE

DDBLTFX ‘‘,**lpDDSZBufferDest**,“‘ Zƒƒƒƒ,Ž— ,, Zƒƒƒƒ ƒƒƒƒ“‘,Ž ,, “ ‘‘,**dwZBufferOpCode** ƒƒƒ,“‘ Zƒƒƒƒ ƒƒƒ ƒ,,,

DDBLT_ZBUFFERSRCCONSTOVERRIDE

DDBLTFX ‘‘, **dwZSrcConst**,“‘ŒZƒƒƒƒ,Ž— ,, Zƒƒƒƒ ƒƒƒƒ“‘,Ž ,, “ ‘‘,**dwZBufferOpCode** ƒƒƒ,“‘ŒZƒƒƒƒ ƒƒƒ ƒ,,,

DDBLT_ZBUFFERSRCOVERRIDE

DDBLTFX ‘‘, **lpDDSZBufferSrc**,“‘ŒZƒƒƒƒ,Ž— ,, Zƒƒƒƒ ƒƒƒƒ“‘,Ž ,, “ ‘‘, **dwZBufferOpCode** ƒƒƒ,“‘ŒZƒƒƒƒ ƒƒƒ ƒ,,,

lpDDBltFx

’‰,,,ƒƒƒƒ“‘Œ‰,Ž’,, **DDBLTFX** ‘‘,ƒƒƒƒ

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 lpDDSAlphaDest;
};
    DWORD dwAlphaSrcConstBitDepth;
union
{
        DWORD               dwAlphaSrcConst;
        LPDIRECTDRAWSURFACE lpDDSAlphaSrc;
};
union
{
        DWORD               dwFillColor;
        DWORD               dwFillDepth;
        LPDIRECTDRAWSURFACE lpDDSPattern;
};
DDCOLORKEY ddckDestColorkey;
DDCOLORKEY ddckSrcColorkey;
} DDBLTFX,FAR* LPDDBLTFX;
```

IDirectDrawSurface2::Blt
IDirectDrawSurface2::BltBatch
DDBLTBATCH の

dwSize

‘‘,‘,, ,,ƒƒƒ, ‘‘,Ž,,,‘, ’,,,,,,,,,,,,

dwDDFX

Œ‰ —,ƒƒƒ,Ž’,,ƒƒƒ

DDBLTFX_ARITHSTRETCHY

,,ƒƒƒƒ“‘,,,,,Ž,‰,,‰Ž ’,Ž—,,

DDBLTFX_MIRRORLEFTRIGHT

,Ž ,ƒ ƒƒƒ,‰“,, ,,ƒƒƒƒ“‘,ƒ ƒƒƒ, ,,‰,“Ž,,

DDBLTFX_MIRRORUPDOWN

,Ž ,ƒ ƒƒƒ,‰“,, ,,ƒƒƒƒ“‘,ƒ ƒƒƒ, ,,‰,“Ž,,

DDBLTFX_NOTEARING

,,ƒƒƒƒ“‘,,,,ƒƒƒƒƒƒ,”,,,,, —,,

DDBLTFX_ROTATE180

,,ƒƒƒƒ“‘,Š,ƒ ƒƒƒ,ŽŒ‰,,180“‰“,,,

DDBLTFX_ROTATE270

,,ƒƒƒƒ“‘,Š,ƒ ƒƒƒ,ŽŒ‰,,270“‰“,,,

DDBLTFX_ROTATE90

,,ƒƒƒƒ“‘,Š,ƒ ƒƒƒ,ŽŒ‰,,90“‰“,,,

DDBLTFX_ZBUFFERBASEDEST

,,ƒƒƒƒ“‘,Š, ”Š,,‘,ƒƒƒƒ“‘Œ,’,,,,ƒƒƒƒ“‘,,ƒƒƒƒ“‘ ,’,,,,,,**dwZBufferBaseDest** ƒƒƒ,‰,,

DDBLTFX_ZBUFFERRANGE

,,ƒƒƒƒ“‘,Š,ƒƒƒƒ“‘Œƒ ƒƒƒ,,• ,,ƒƒƒ,ŒŠ,Ž’,,”ˆ’,,,**dwZBufferLow**,,, **dwZBufferHigh** ƒƒƒ,Ž—,,

dwROP

Win32 ƒƒƒ —

dwDDROP

DirectDraw ƒƒƒ —

dwRotationAngle

ƒƒƒƒ“‘,‰“Š“

dwZBufferOpCode

Zƒƒƒƒ”Š

dwZBufferLow

Zƒƒƒƒ, ’ŒŠ

dwZBufferHigh

Zƒƒƒƒ, ŒŠ

dwZBufferBaseDest

Zƒƒƒƒ,ƒƒƒƒ“‘ Š’

dwZDestConstBitDepth

ƒƒƒƒ“‘ ,’ ,ƒƒƒ “

dwZDestConst

ƒƒƒƒ“‘ Zƒƒƒƒ,,,Ž—,,,’

lpDDSZBufferDest

ƒƒƒƒ“‘ Zƒƒƒƒ,,,Ž—,,,ƒ ƒƒƒ

dwZSrcConstBitDepth

ƒƒƒƒ“‘Œ,’ ,ƒƒƒ “

dwZSrcConst

ƒƒƒƒ“‘ŒZƒƒƒƒ,,,Ž—,,,’

lpDDSZBufferSrc
ƒƒƒƒ“‘ŒZƒƒƒƒ‚‚‚Ž—‚‚‚ƒ ƒƒƒ

dwAlphaEdgeBlendBitDepth
ƒƒƒƒ ƒƒƒ ‚‚‚‚ƒƒƒ “’

dwAlphaEdgeBlend
ƒƒƒ ‚‚‚‚Ž‚‚‚ƒƒƒƒ’

dwReserved
—‚‚‚‚—‚‚‚‚‚

dwAlphaDestConstBitDepth
ƒƒƒƒ“‘ ƒƒƒƒ’ ‚ƒƒƒ “

dwAlphaDestConst
ƒƒƒƒ“‘ ƒƒƒƒ ƒƒƒƒ‚‚‚‚Ž‚‚‚’

lpDDSAlphaDest
ƒƒƒƒ“‘ ƒƒƒƒ ƒƒƒƒ‚‚‚Ž—‚‚‚ƒ ƒƒƒ

dwAlphaSrcConstBitDepth
ƒƒƒƒ“‘Œƒƒƒƒ’ ‚ƒƒƒ “

dwAlphaSrcConst
ƒƒƒƒ“‘Œƒƒƒƒ ƒƒƒƒ‚‚‚‚Ž‚‚‚’

lpDDSAlphaSrc
ƒƒƒƒ“‘Œƒƒƒƒ ƒƒƒƒ‚‚‚Ž—‚‚‚ƒ ƒƒƒ

dwFillColor
DDBLT_COLORFILL ‚Ž’‚‚‚Ž ƒ ƒƒƒ‚• ‚‚‚‚‚Ž—
‚‚‚ ‚Ž’ ‚‚’‚ƒ ƒƒƒ ƒƒƒ‚ˆ‘‚‚RGB‚ ’‚‚‚‚ƒƒƒƒ ƒƒƒƒƒƒ‚‚‚

dwFillDepth
Zƒƒƒƒ‚ “’

lpDDSPattern
ƒƒ ƒ‚‚‚Ž—‚‚ƒ ƒƒƒ ƒƒ ƒ‚ƒƒƒƒ“‘Œ‚ƒƒƒƒ“‘ ‚ƒƒƒƒ‚‚“’‚ƒƒƒƒ“‘ ‚Ž—‚‚‚

ddckDestColorkey
— ‚‚ƒƒƒƒ“‘ ƒƒ ƒ

ddckSrcColorkey
— ‚‚ƒƒƒƒ“‘Œƒƒ ƒ

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;

} DDCAPS,FAR* LPDDCAPS;
```

DirectDraw のの DirectDrawSurface **DDSCAPS**

IDirectDraw2::GetCaps **IDirectDrawPalette::GetCaps**

dwSize

‘‘,‘,, ,,ƒƒƒ, ‘‘,Ž,,,‘, ’,,,,,,,,,,

dwCaps

ƒƒƒƒ“’”—,,,,,ƒƒƒ

DDCAPS_3D

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,,,ƒƒƒƒƒ ƒƒƒ,Ž,,,,,,,•,

DDCAPS_ALIGNBOUNDARYDEST

DirectDraw,ƒƒƒƒ“‘Œ‹Œ, •, **dwAlignBoundaryDest**
,” ,,, ,,,ƒƒ ƒ,,,,,•,

DDCAPS_ALIGNBOUNDARYSRC

DirectDraw,ƒƒƒƒ“‘Œ‹Œ, •, **dwAlignBoundarySrc**
,” ,,, ,,,ƒƒ ƒ,,,,,•,

DDCAPS_ALIGNSIZEDEST

DirectDraw
,ƒƒƒ’ˆ,,,Ž•Œ,‘,,,**dwAlignSizeDest**,” ,,, ,ƒƒƒƒ“‘Œ‹Œ,,,ƒƒ ƒ,,,,,•,

DDCAPS_ALIGNSIZESRC

DirectDraw
,ƒƒƒ’ˆ,,,•Œ,‘,,,**dwAlignSizeSrc**,” ,,, ,ƒƒƒƒ“‘Œ‹Œ,,,ƒƒ ƒ,,,,,•,

DDCAPS_ALIGNSTRIDE

DirectDraw ,**dwAlignStrideAlign**
’,“,,ƒƒƒƒƒ‹Š,Ž,,,,ƒƒƒƒƒƒ ƒƒƒ ƒ ƒƒƒ, ,,,,,•,

DDCAPS_ALPHA

ƒƒƒƒ“‘ ,Š ƒƒƒƒƒƒ ƒ ƒƒƒƒ,ƒƒƒƒ ƒƒƒƒ,ƒƒ ƒ,,,,,•,

DDCAPS_BANKSWITCHED

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,ƒƒƒ ,‘,,,, ƒƒƒƒƒƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ,” ,’,‰” ,,,,,,
•,

DDCAPS_BLT

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,ƒƒƒƒ“‘ ,,,,,,,•,

DDCAPS_BLTCOLORFILL

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,ƒƒƒƒ“‘,ƒƒ ƒƒƒ,,,,,,,•,

DDCAPS_BLTDEPTHFILL

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,ƒƒƒƒ“‘, Zƒƒƒƒ “• ,,,,,,,•,

DDCAPS_BLTFOURCC

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,ƒƒƒƒ“‘, ƒƒ ƒƒ ƒ•Š,,,,,,,•,

DDCAPS_BLTQUEUE

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,”“Šƒƒƒƒ“‘,,,,,,,•,

DDCAPS_BLTSTRETCH

ƒƒƒƒ“‘ , ƒƒƒƒƒƒ ƒ ƒƒƒƒ, ’,,,,,,•,

DDCAPS_CANBLTSYSMEM

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,ƒƒƒƒ ƒƒƒ,,,,,ƒƒƒƒ ƒƒƒ,,ƒƒƒƒ“‘,,,,,,•,

DDCAPS_CANCLIP

ƒƒƒƒƒƒ ƒ ƒƒƒƒ,ƒƒƒƒ“‘,ƒƒƒƒƒƒ,,,,,,•,

DDCAPS_CANCLIPSTRETCHED

ƒƒƒƒƒƒ ƒ ƒƒƒƒ‚ ’ƒƒƒƒ“‘‚ƒƒƒƒƒƒ‚‚‚‚‚•‚

DDCAPS_COLORKEY

ƒ ƒ ƒƒ‚‚‚ƒƒƒƒ“‘ ‚‚‚‚‚ƒƒ ƒ ‚‚‚‚‚ŒŽ‚ƒƒ ƒ‚‚ “’‚ƒƒ ƒ ”— •‚ **dwCKeyCaps** ƒƒƒ‚‚‚

DDCAPS_COLORKEYHWASSIST

ƒƒ ƒ ‚ƒ ƒƒƒƒ‚‚‚ƒƒ ƒ‚‚‚‚‚‚‚•‚

DDCAPS_GDI

ƒƒƒƒƒƒ ƒ ƒƒƒƒ‚ GDI ‚‹—‚‚‚‚‚‚•‚

DDCAPS_NOHARDWARE

ƒ ƒƒƒƒ ƒƒ ƒ‚‚‚‚‚‚•‚

DDCAPS_OVERLAY

ƒƒƒƒƒƒ ƒ ƒƒƒƒ‚ƒ ƒ ƒƒ‚ƒƒ ƒ‚‚‚‚‚•‚

DDCAPS_OVERLAYCANTCLIP

ƒƒƒƒƒƒ ƒ ƒƒƒƒ‚ƒ ƒ ƒƒ‚ƒƒ ƒ‚‚‚ ƒƒƒƒƒƒ‚‚‚‚‚‚‚‚•‚

DDCAPS_OVERLAYFOURCC

ƒ ƒ ƒƒ ƒ ƒƒƒƒ‚ƒ ƒ ƒƒ —‚Š ƒƒ ƒƒ ƒ•Š‚‚‚‚‚‚‚•‚

DDCAPS_OVERLAYSTRETCH

ƒ ƒ ƒƒ ƒ ƒƒƒƒ‚ ’‚”—‚‚‚‚‚‚•‚

DDCAPS_PALETTE

DirectDraw ‚ƒƒƒƒƒ ƒ ƒƒƒ‚‚‚‚‚ ‘‚ƒ ƒƒƒ‚‚ DirectDrawPalette ƒƒƒƒƒƒ‚ ‚‚ ƒƒ ƒ‚‚‚‚‚‚Ž‚

DDCAPS_PALETTEVSYNC

DirectDraw‚VSYNC‚“Š‚‚ƒƒƒƒ‚ ‚‚‚‚‚‚•‚

DDCAPS_READSCANLINE

ƒƒƒƒƒƒ ƒ ƒƒƒƒ‚ƒƒƒƒƒƒƒ“‚Ž‚‚‰”‚‚‚‚‚‚•‚

DDCAPS_STEREOVIEW

ƒƒƒƒƒƒ ƒ ƒƒƒƒ‚ƒƒƒƒ ƒƒƒƒ”—‚Ž‚‚‚‚‚‚‚•‚

DDCAPS_VBI

ƒƒƒƒƒƒ ƒ ƒƒƒƒ‚VƒƒƒƒŠ‚ ‚‚ ‚‚‚‚‚‚•‚

DDCAPS_ZBLTS

ƒƒƒƒ“‘ ‚‚‚‚Zƒƒƒƒ‚Ž—‚ƒƒ ƒ‚‚

DDCAPS_ZOVERLAYS

‚‚‚ Œ‚‚ƒ ƒ ƒƒ‚’‚‚‚**IDirectDrawSurface2::UpdateOverlayZOrder**ƒ ƒƒƒ‚Ž—‚ƒƒ ƒ‚‚

dwCaps2

‚‚ƒƒƒƒ“’”—‚ ‚•‚ƒƒƒ

DDCAPS2_CERTIFIED

ƒƒƒƒƒƒ ƒ ƒƒƒƒ‚• ‚‚‚‚‚‚‚‚•‚

DDCAPS2_NO2DDURING3DSCENE

IDirect3DDevice::BeginSceneƒƒƒƒŒ‚ ‚‚ **IDirect3DDevice::EndScene** ƒƒƒƒŒ‚ ‚‚‚Š‚ Direct3D‚Ž—

,,,,,,,ƒ ƒƒƒ , **IDirectDrawSurface2::Blt**,,**IDirectDrawSurface2::Lock**,,,, 2D —,Ž ,,,,,,•,

dwCKeyCaps

ƒƒ ƒ ”—,•,ƒƒƒ

DDCKEYCAPS_DESTBLT

ƒƒ ƒ ,,,“‰Œƒƒƒƒ“‘,ƒƒ ƒ,, ƒƒ ƒ ,RGB ,ƒƒƒƒ“‘ ƒ ƒƒƒ,’Š‰”,ƒƒƒ,Ž,

DDCKEYCAPS_DESTBLTCLRSPACE

ƒƒ ƒƒ ƒ,,,“‰Œƒƒƒƒ“‘,ƒƒ ƒ,, ƒƒ ƒƒ ƒ,RGB ,ƒƒƒƒ“‘ ƒ ƒƒƒ,’Š‰”,ƒƒƒ,Ž,

DDCKEYCAPS_DESTBLTCLRSPACEYUV

ƒƒ ƒƒ ƒ,,,“‰Œƒƒƒƒ“‘,ƒƒ ƒ,, ƒƒ ƒƒ ƒ,YUV ,ƒƒƒƒ“‘ ƒ ƒƒƒ,’Š‰”,ƒƒƒ,Ž,

DDCKEYCAPS_DESTBLTYUV

ƒƒ ƒ ,,,“‰Œƒƒƒƒ“‘,ƒƒ ƒ,, ƒƒ ƒ ,YUV ,ƒƒƒƒ“‘ ƒ ƒƒƒ,’Š‰”,ƒƒƒ,Ž,

DDCKEYCAPS_DESTOVERLAY

ƒƒ ƒ ƒƒƒ,,,ƒ ƒ ƒƒ,ƒƒ ƒ,, ƒƒ ƒ ,RGB ,ƒ ƒ ƒƒ,,,ƒƒƒƒ“‘ ƒ ƒƒƒ,’Š‰”,ƒƒƒ

DDCKEYCAPS_DESTOVERLAYCLRSPACE

RGB ,ƒƒƒƒ“‘ ƒƒ ƒ ,,,ƒƒ ƒƒ ƒ,ƒƒ ƒ,,

DDCKEYCAPS_DESTOVERLAYCLRSPACEYUV

YUV ,ƒƒƒƒ“‘ ƒƒ ƒ ,,,ƒƒ ƒƒ ƒ,ƒƒ ƒ,,

DDCKEYCAPS_DESTOVERLAYONEACTIVE

‰Ž ‘,ƒ ƒ ƒƒ ƒ ƒƒƒ, ,,,,,,—Œ,ƒƒƒƒ“‘ ƒƒ ƒ ’,ƒƒ ƒ,,

DDCKEYCAPS_DESTOVERLAYYUV

ƒƒ ƒ ƒƒƒ,,,ƒ ƒ ƒƒ,ƒƒ ƒ,, ƒƒ ƒ , YUV ,ƒ ƒ ƒƒ,,,ƒƒƒƒ“‘ ƒ ƒƒƒ,’Š‰”,ƒƒƒ

DDCKEYCAPS_NOCOSTOVERLAY

ƒ ƒ ƒƒ,ƒƒ ƒ ,Ž—,,,,,‘,, —‘“,‰‹,,,,,,•,

DDCKEYCAPS_SRCBLT

RGB ƒ ƒƒƒ,ƒƒƒƒ“‘Œƒƒ ƒ ,,,“‰Œƒƒƒƒ“‘,ƒƒ ƒ,,

DDCKEYCAPS_SRCBLTCLRSPACE

RGB ƒ ƒƒƒ,ƒƒƒƒ“‘Œƒƒ ƒƒ ƒ,,,“‰Œƒƒƒƒ“‘,ƒƒ ƒ,,

DDCKEYCAPS_SRCBLTCLRSPACEYUV

YUV ƒ ƒƒƒ,ƒƒƒƒ“‘Œƒƒ ƒƒ ƒ,,,“‰Œƒƒƒƒ“‘,ƒƒ ƒ,,

DDCKEYCAPS_SRCBLTYUV

YUV ƒ ƒƒƒ,ƒƒ ƒ ,,,“‰Œƒƒƒƒ“‘,ƒƒ ƒ,,

DDCKEYCAPS_SRCOVERLAY

RGB ,ƒƒƒƒ“‘Œƒƒ ƒ ,,,ƒ ƒ ƒƒ,ƒƒ ƒ,,

DDCKEYCAPS_SRCOVERLAYCLRSPACE

RGB ƒ ƒ ƒƒ ƒ ƒƒƒ‚ƒƒƒƒ“‘Œƒƒ ƒ ‚‚‚ƒƒ ƒƒ ƒ‚Ž‚‚ƒ ƒ ƒƒ‚ƒƒ ƒ‚ ‚

DDCKEYCAPS_SRCOVERLAYCLRSPACEYUV

YUV ƒ ƒ ƒƒ ƒ ƒƒƒ‚ƒƒƒƒ“‘Œƒƒ ƒ ‚‚‚ƒƒ ƒƒ ƒ‚ƒƒ ƒ‚‚

DDCKEYCAPS_SRCOVERLAYONEACTIVE

‰Ž ‘‚ƒ ƒ ƒƒ ƒ ƒƒƒ‚ ‚‚‚‚‚‚—Œ‚ƒƒƒƒ“‘Œƒƒ ƒ ’‚ƒƒ ƒ‚‚

DDCKEYCAPS_SRCOVERLAYYUV

YUV ƒ ƒ ƒƒ ƒ ƒƒƒ‚ƒƒƒƒ“‘Œƒƒ ƒ ‚‚‚ƒ ƒ ƒƒ‚ƒƒ ƒ‚‚

dwFXCaps

ƒƒƒƒ“’‚ ’‚‚‚FX”—

DDFXCAPS_BLTARITHSTRETCHY

ƒƒƒƒ‚‚ ‚‚‚Š‘‚‚ƒƒƒƒƒ‚‚‚ ƒƒƒƒ“‘‘ ‚Š‚Š‘‚ ƒ ƒƒƒ‚ ‚‚‰Ž —‚Ž— ‚‚ ‚‚‚ ‚Ž ’‚ ‚‰‚‚ ‚‚‚

DDFXCAPS_BLTARITHSTRETCHYN

ƒƒƒƒ‚‚ ‚‚‚Š‘‚‚ƒƒƒƒƒ‚‚‚ ƒƒƒƒ“‘‘ ‚Š‚Š‘‚‚ ƒ ƒƒƒ‚ ‚‚‰Ž —‚Ž— ‚‚ ‚‚‚‚Ž ’‚ ‚‰‚‚ ‚‚ ” ×1, ×2, ‚‚‚‚ ‚Š‘‚‚‚‚‚“ ‚‚

DDFXCAPS_BLTMIRRORLEFTRIGHT

ƒƒƒƒ“‘‘ ’‚ ‰”“‚‚‚‚‚ƒƒ ƒ‚‚

DDFXCAPS_BLTMIRRORUPDOWN

ƒƒƒƒ“‘‘ ’‚ ‰”“‚‚‚‚‚ƒƒ ƒ‚‚

DDFXCAPS_BLTROTATION

ƒƒƒƒ“‘‘ ’‚”ˆ‚‰“‚ƒƒ ƒ‚‚

DDFXCAPS_BLTROTATION90

ƒƒƒƒ“‘‘ ’‚‚‚“‚‰“‚ƒƒ ƒ‚‚

DDFXCAPS_BLTSHRINKX

‚Ž • ‚‰‚‚ƒ ƒƒƒ‚”ˆ‚ ‚‚‚‚‚ƒƒ ƒ‚‚ ‚‚ƒƒƒ‚ƒƒƒƒ“‘‘ ‚‚‚—Œ‚‚‚

DDFXCAPS_BLTSHRINKXN

‚Ž • ‚‰‚‚ƒ ƒƒƒ‚ •‚ˆ ×1/2 ×1/3‚‚‚‚ ‚ ‚‚‚‚‚ƒƒ ƒ‚‚ ‚‚ƒƒƒ‚ƒƒƒƒ“‘‘ ‚‚‚—Œ‚‚‚

DDFXCAPS_BLTSHRINKY

‚Ž ’ ‚‰‚‚ƒ ƒƒƒ‚”ˆ‚ ‚‚‚‚‚ƒƒ ƒ‚‚ ‚‚ƒƒƒ‚ƒƒƒƒ“‘‘ ‚‚‚—Œ‚‚‚

DDFXCAPS_BLTSHRINKYN

‚Ž ’ ‚‰‚‚ƒ ƒƒƒ‚ •‚ˆ ×1/2 ×1/3‚‚‚‚ ‚ ‚‚‚‚‚ƒƒ ƒ‚‚ ‚‚ƒƒƒ‚ƒƒƒƒ“‘‘ ‚‚‚—Œ‚‚‚

DDFXCAPS_BLTSTRETCHX

‚Ž • ‚‰‚‚ƒ ƒƒƒ‚”ˆ‚Š‘‚‚‚‚‚ƒƒ ƒ‚‚ ‚‚ƒƒƒ‚ƒƒƒƒ“‘‘ ‚‚‚—Œ‚‚‚

DDFXCAPS_BLTSTRETCHXN

‚Ž • ‚‰‚‚ƒ ƒƒƒ‚ ” ×1, ×2, ‚‚‚‚ ‚Š‘‚‚‚‚‚ƒƒ ƒ‚‚ ‚‚ƒƒƒ‚ƒƒƒƒ“‘‘ ‚‚‚—Œ‚‚‚

DDFXCAPS_BLTSTRETCHY

,Ž ’ ,‰,,ƒ ƒƒƒ,”ˆ,Š‘,,,,,ƒƒ ƒ,, ,,ƒƒƒ,ƒƒƒƒ“‘ ,,,—Œ,,,

DDFXCAPS_BLTSTRETCHYN

,Ž ’ ,‰,,ƒ ƒƒƒ, ” ×1, ×2, ,,,, ,Š‘,,,,,ƒƒ ƒ,, ,,ƒƒƒ,ƒƒƒƒ“‘ ,,,—Œ,,,

DDFXCAPS_OVERLAYARITHSTRETCHY

ƒƒƒƒ,, ,,,Š‘,,ƒƒƒƒƒ,,,, ƒƒƒƒ“‘ ,Š,Š‘, ƒ ƒƒƒ, ,,‰Ž —,Ž— ,, ,Ž ’ ,‰,, ,,,

DDFXCAPS_OVERLAYARITHSTRETCHYN

ƒƒƒƒ,, ,,,Š‘,,ƒƒƒƒƒ,,,, ƒƒƒƒ“‘ ,Š,Š‘, ƒ ƒƒƒ, ,,‰Ž —,Ž— ,, ,Ž ’ ,‰,, ,, ” ×1, ×2, ,,,, ,Š‘, ,,,

DDFXCAPS_OVERLAYMIRRORLEFTRIGHT

‰”“ƒ ƒ ƒƒ,ƒƒ ƒ,,

DDFXCAPS_OVERLAYMIRRORUPDOWN

‰”“ƒ ƒ ƒƒ,ƒƒ ƒ,,

DDFXCAPS_OVERLAYSHRINKX

,Ž • ,‰,,ƒ ƒƒƒ,”ˆ, ,,,,,ƒƒ ƒ,, ,,ƒƒƒ, DDSCAPS_OVERLAY ƒ ƒƒƒ,,,—Œ,,, ,,ƒƒƒ,ƒ ƒƒƒ,”—,,,Ž, ,‰”,,,,,,•,,,

DDFXCAPS_OVERLAYSHRINKXN

,Ž • ,‰,,ƒ ƒƒƒ, ” ×1, ×2, ,,,, , ,,,,,ƒƒ ƒ,, ,,ƒƒƒ, DDSCAPS_OVERLAY ƒ ƒƒƒ,,,—Œ,,, ,,ƒƒƒ,ƒ ƒƒƒ,”— ,,,Ž, ,‰”,,,,,,•,,,

DDFXCAPS_OVERLAYSHRINKY

,Ž ’ ,‰,,ƒ ƒƒƒ,”ˆ, ,,,,,ƒƒ ƒ,, ,,ƒƒƒ, DDSCAPS_OVERLAY ƒ ƒƒƒ,,,—Œ,,, ,,ƒƒƒ,ƒ ƒƒƒ,”—,,,Ž, ,‰”,,,,,,•,,,

DDFXCAPS_OVERLAYSHRINKYN

,Ž ’ ,‰,,ƒ ƒƒƒ, •,ˆ ×1/2 ×1/3,,,, , ,,,,,ƒƒ ƒ,, ,,ƒƒƒ, DDSCAPS_OVERLAY ƒ ƒƒƒ,,,—Œ,,, ,,ƒƒƒ,ƒ ƒƒƒ,”— ,,,Ž, ,‰”,,,,,,•,,,

DDFXCAPS_OVERLAYSTRETCHX

,Ž • ,‰,,ƒ ƒƒƒ,”ˆ,Š‘,,,,,ƒƒ ƒ,, ,,ƒƒƒ, DDSCAPS_OVERLAY ƒ ƒƒƒ,,,—Œ,,, ,,ƒƒƒ,ƒ ƒƒƒ,”—,,,Ž, ,‰”,,,,,,•,,,

DDFXCAPS_OVERLAYSTRETCHXN

,Ž • ,‰,,ƒ ƒƒƒ, ” ×1, ×2, ,,,, ,Š‘,,,,,ƒƒ ƒ,, ,,ƒƒƒ, DDSCAPS_OVERLAY ƒ ƒƒƒ,,,—Œ,,, ,,ƒƒƒ,ƒ ƒƒƒ,”— ,,,Ž, ,‰”,,,,,,•,,,

DDFXCAPS_OVERLAYSTRETCHY

,Ž ’ ,‰,,ƒ ƒƒƒ,”ˆ,Š‘,,,,,ƒƒ ƒ,, ,,ƒƒƒ,DDSCAPS_OVERLAY ƒ ƒƒƒ,,,—Œ,,, ,,ƒƒƒ,ƒ ƒƒƒ,”—,,,Ž, ,‰”,,,,,,•,,,

DDFXCAPS_OVERLAYSTRETCHYN

,Ž ’ ,‰,,ƒ ƒƒƒ, ” ×1, ×2, ,,,, ,Š‘,ƒƒ ƒ,, ,,ƒƒƒ,DDSCAPS_OVERLAY ƒ ƒƒƒ,,,—Œ,,, ,,ƒƒƒ,ƒ ƒƒƒ,”— ,,,Ž, ,‰”,,,,,,•,,,

dwFXAlphaCaps

ƒƒƒƒ“’,ƒƒƒƒ”—

DDFXALPHACAPS_BLTALPHAEDGEBLEND

ƒƒƒƒ“‘Œƒƒ ƒ ,ƒ ƒƒƒ,ƒƒƒŽ,,ƒƒƒƒ ƒƒƒƒƒƒƒ,ƒƒ ƒ,, ƒƒƒƒ“‘‘ ,,,, Ž—,,,

DDFXALPHACAPS_BLTALPHAPIXELS

ƒƒƒƒ ƒƒ ƒƒƒ,ƒƒƒƒ •,ƒƒ ƒ,,
ƒƒƒƒ ƒƒ ƒƒƒ,,ƒƒƒƒ •,ƒƒƒ “,, , , ,,,,,,,, ƒƒƒƒ’,‘‰,,,,,,•“–,,,
ƒƒƒƒ •, “,,,,,, ,,,,,Š‘,“–,’,,, ƒƒƒƒ“‘‘ ,,,,Ž—,,,

DDFXALPHACAPS_BLTALPHAPIXELSNEG

ƒƒƒƒ ƒƒ ƒƒƒ,ƒƒƒƒ •,ƒƒ ƒ,,
ƒƒƒƒ ƒƒ ƒƒƒ,,ƒƒƒƒ •,ƒƒƒ “,, , , ,,,,,,,, ƒƒƒƒ’,‘‰,,,,,,“–,,,
ƒƒƒƒ •, “,,,,,, ,,,,,Š‘,•“–,’,,, ,,ƒƒƒ, DDCAPS_ALPHA ,ƒƒƒ,,, ,Œ,Ž,,,,,,, ƒƒƒƒ“‘‘ ,,,,Ž—,,,

DDFXALPHACAPS_BLTALPHASURFACES

ƒƒƒƒ —ƒ ƒƒƒ,ƒƒ ƒ,, ƒƒƒƒ —ƒ ƒƒƒ,ƒƒƒ “,, , , ,,,,,,,,“,
ƒƒƒƒ’,‘‰,,,,,,•“–,,, ƒƒƒƒ •, “,,,,,, ,,,,,Š‘,“–,’,,, ƒƒƒƒ“‘‘ ,,,,Ž —,,,

DDFXALPHACAPS_BLTALPHASURFACESNEG

ƒƒƒƒ ƒƒƒƒ, ƒƒƒƒ’,‘‰,,,,“–,,,,,,•,
ƒƒƒƒ ƒƒƒƒ ƒ ƒ, “,, , , ,,,,,,,, ƒƒƒƒ •, “,,,,,, ,,,,,Š‘,•“– ,’,,, ,,ƒƒƒ, DDCAPS_ ALPHASURFACES ,ƒƒƒ,,, ,Œ,ƒƒƒ,,,,,,,, ƒƒƒƒ“‘‘ ,,,,Ž—,,,

DDFXALPHACAPS_OVERLAYALPHAEDGEBLEND

ƒƒƒƒ“‘Œƒƒ ƒ ,ƒ ƒƒƒ,ƒƒƒŽ,,ƒƒƒƒ ƒƒƒƒƒƒƒ,ƒƒ ƒ,, ƒ ƒ ƒƒ,,,,Ž —,,,

DDFXALPHACAPS_OVERLAYALPHAPIXELS

ƒƒƒƒ ƒƒ ƒƒƒ,ƒƒƒƒ •,ƒƒ ƒ,,
ƒƒƒƒ ƒƒ ƒƒƒ,,ƒƒƒƒ •,ƒƒƒ “,, , , ,,,,,,,, ƒƒƒƒ’,‘‰,,,,,,•“–,,,
ƒƒƒƒ •, “,,,,,, ,,,,,Š‘,“–,’,,, ƒ ƒ ƒƒ,,,,Ž—,,,

DDFXALPHACAPS_OVERLAYALPHAPIXELSNEG

ƒƒƒƒ ƒƒ ƒƒƒ,ƒƒƒƒ •,ƒƒ ƒ,,
ƒƒƒƒ ƒƒ ƒƒƒ,,ƒƒƒƒ •,ƒƒƒ “,, , , ,,,,,,,, ƒƒƒƒ’,‘‰,,,,,,“–,,,
ƒƒƒƒ •, “,,,,,, ,,,,,Š‘,•“–,’,,, ,,ƒƒƒ, DDCAPS_ ALPHAPIXELS ,ƒƒƒ,,, ,Œ,Ž,,,,,,, ƒ ƒ ƒƒ,,,,Ž—,,,

DDFXALPHACAPS_OVERLAYALPHASURFACES

ƒƒƒƒ —ƒ ƒƒƒ,ƒƒ ƒ,, ƒƒƒƒ —ƒ ƒƒƒ,ƒƒƒ “,, , , ,,,,,,,,
ƒƒƒƒ’,‘‰,,,,,,•“–,,, ƒƒƒƒ •, “,,,,,, ,,,,,Š‘,“–,’,,, ƒ ƒ ƒƒ,,,,Ž —,,,

DDFXALPHACAPS_OVERLAYALPHASURFACESNEG

ƒƒƒƒ ƒƒƒƒ, ƒƒƒƒ’,‘‰,,,,“–,,,,,,•,
ƒƒƒƒ ƒƒƒƒ ƒ ƒ, “,, , , ,,,,,,,, ƒƒƒƒ •, “,,,,,, ,,,,,Š‘,•“– ,’,,, ,,ƒƒƒ,DDCAPS_ ALPHASURFACES ,ƒƒƒ,,, ,Œ,ƒƒƒ,,,,,,,, ƒ ƒ ƒƒ,,,,Ž—,,,

dwPalCaps

ƒƒƒƒ”—,•,ƒƒƒ

DDPCAPS_1BIT

ƒƒƒƒƒƒ,,ƒƒƒ,,,,,,•, ƒƒ ƒ ƒƒ,2,, –,,,

DDPCAPS_2BIT

ƒƒƒƒƒƒ,,ƒƒƒ,,,,,,•, ƒƒ ƒ ƒƒ,4,, –,,,

DDPCAPS_4BIT

ƒƒƒƒƒƒ,,ƒƒƒ,,,,,,•, ƒƒ ƒ ƒƒ,16, –,,,

DDPCAPS_8BIT

ƒƒƒƒƒƒ,,ƒƒƒ,,,,,,•, ƒƒ ƒ ƒƒ,256, –,,,

DDPCAPS_8BITENTRIES

ƒƒƒƒƒƒ,,ƒƒƒ,ƒƒ ƒƒƒƒƒƒ,Ž',, ,,ƒƒ ƒƒ, ,,DDPCAPS_1BIT DDPCAPS_2BIT ,,,, DDPCAPS_4BIT"—,Ž,,,Ž,, ,,,– •ƒ ƒƒƒ,ƒƒƒƒ bpp –,,ƒƒƒ,,, ,—Œ,,, ,,,,, – ,',,ƒƒƒ,,, ƒƒƒƒ"' ƒ ƒƒƒ ,, bpp ƒƒƒƒ,ƒƒƒƒƒƒ,,,

DDPCAPS_ALLOW256

,,ƒƒƒƒ,256, –,,,,'‹,,,,,,Ž,

DDPCAPS_PRIMARYSURFACE

ƒƒƒƒ,ƒƒƒƒƒ ƒ ƒƒƒ,ƒƒƒƒ,,,,,,,,•, ƒƒƒƒ,• , DDPCAPS_VSYNC "—,Ž' ƒƒ ƒ,,,,,, ƒƒƒƒƒƒ,' ,"‰,,,

DDPCAPS_PRIMARYSURFACELEFT

ƒƒƒƒ, 'ƒƒƒƒƒ ƒ ƒƒƒ,ƒƒƒƒ,,,,,,•, ƒƒƒƒ,• , DDPCAPS_VSYNC"—,Ž' ƒƒ ƒ,,,,,, ƒƒƒƒƒƒ,' ,"‰,,,

DDPCAPS_VSYNC

ƒƒƒƒ,ƒƒƒ ,ƒƒƒƒƒƒ,"Š,, ,,,,,,•,

dwSVCaps

ƒƒƒƒ ƒƒƒƒ"—,•,ƒƒƒ

DDSVCAPS_ENIGMA

ƒƒƒƒ ƒ ƒƒƒƒ,,,ƒƒƒƒ ƒƒ ,ŽŒ,,,,,,,•,

DDSVCAPS_FLICKER

Ž"ƒƒƒƒƒƒƒ,,,ƒƒƒƒ ƒƒ ,ŽŒ,,,,,,,•,

DDSVCAPS_REDBLUE

, ,ƒƒƒƒ, ,‰,Š,",,3DŠ‹,ƒƒƒƒ ƒƒ ,ŽŒ,,,,,,,•, ,,,,‰'(,ƒƒ ƒƒ ƒ),,,Ž',," ,,,,,,,,,,

DDSVCAPS_SPLIT

•Šƒƒƒ ƒ,,,ƒƒƒƒ ƒƒ ,ŽŒ,,,,,,,•,

dwAlphaBltConstBitDepths

DDBD_2 DDBD_4,,, DDBD_8 ƒƒƒƒ–,2ƒƒƒ 4ƒƒƒ ,,,,8ƒƒƒ,Ž,

dwAlphaBltPixelBitDepths

DDBD_1 DDBD_2 DDBD_4,,, DDBD_8 ƒƒƒƒ–,1ƒƒƒ 2ƒƒƒ 4ƒƒƒ ,,,,8ƒƒƒ,Ž,

dwAlphaBltSurfaceBitDepths

DDBD_1 DDBD_2 DDBD_4,,, DDBD_8 ƒƒƒƒ–,1ƒƒƒ 2ƒƒƒ 4ƒƒƒ ,,,,8ƒƒƒ,Ž,

dwAlphaOverlayConstBitDepths
DDBD_2 DDBD_4,,, DDBD_8 ƒƒƒƒ–,2ƒƒƒ 4ƒƒƒ ,,,,8ƒƒƒ,Ž,

dwAlphaOverlayPixelBitDepths
DDBD_1 DDBD_2 DDBD_4,,, DDBD_8 ƒƒƒƒ–,1ƒƒƒ 2ƒƒƒ 4ƒƒƒ ,,,,8ƒƒƒ,Ž,

dwAlphaOverlaySurfaceBitDepths
DDBD_1 DDBD_2 DDBD_4,,, DDBD_8 ƒƒƒƒ–,1ƒƒƒ 2ƒƒƒ 4ƒƒƒ ,,,,8ƒƒƒ,Ž,

dwZBufferBitDepths
DDBD_8 DDBD_16 DDBD_24,,, DDBD_32 ƒƒƒƒ– ,1ƒƒƒ 16ƒƒƒ 24ƒƒƒ,,,,32ƒƒƒ,Ž,

dwVidMemTotal
ƒƒƒƒƒƒ ƒƒƒ,‘‘,—

dwVidMemFree
‹,ƒƒƒƒƒƒ ƒƒƒ,—

dwMaxVisibleOverlays
‰Ž ‘,ƒ ƒ ƒƒ, ‘,

dwCurrVisibleOverlays
‰Ž ‘,ƒ ƒ ƒƒ,Œ ,

dwNumFourCCCodes
FourCC ƒ ƒ,

dwAlignBoundarySrc
ƒƒƒƒ“‘Œ‹Œ, •,’ˆ

dwAlignSizeSrc
ƒƒƒƒ“‘Œ‹Œ,‘,,

dwAlignBoundaryDest
ƒƒƒƒ“‘ ‹Œ, •,’ˆ

dwAlignSizeDest
ƒƒƒƒ“‘ ‹Œ,‘,,,’ˆ

dwAlignStrideAlign
ƒƒƒƒƒ —

dwRops[DD_ROP_SPACE]
ƒƒƒ —,ƒƒ ƒ

ddsCaps
ˆ”“,”—,Ž, **DDSCAPS** ‘‘

dwMinOverlayStretch and **dwMaxOverlayStretch**
1000”,,, ‘Œ,ƒ ƒ ƒƒŠ‘Œ ,,,, 1.3=1300

dwMinLiveVideoStretch and **dwMaxLiveVideoStretch**
1000”,,, ‘Œ,ƒƒƒ ƒƒƒŠ‘Œ ,,,, 1.3=1300

dwMinHwCodecStretch and **dwMaxHwCodecStretch**
1000”,,, ‘Œ,ƒ ƒƒƒƒ ƒ ƒƒƒŠ‘Œ ,,,, 1.3=1300

dwReserved1, **dwReserved2**, **dwReserved3**
—,,,,––,,,,,

dwSVBCaps
ƒƒƒƒ ƒƒƒ ƒƒƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒ”—

dwSVBCKeyCaps
ƒƒƒƒ ƒƒƒ ƒƒƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒƒƒ ƒ ŒŠ

dwSVBFXCaps
ƒƒƒƒ ƒƒƒ ƒƒƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒFXŒŠ

dwSVBRops[DD_ROP_SPACE]
ƒƒƒƒ ƒƒƒ ƒƒƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒ —ƒƒ ƒ

dwVSBCaps
ƒƒƒƒƒƒ ƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒ”—

dwVSBCKeyCaps
ƒƒƒƒƒƒ ƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒƒƒ ƒ ŒŠ

dwVSBFXCaps
ƒƒƒƒƒƒ ƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒFXŒŠ

dwVSBRops[DD_ROP_SPACE]
ƒƒƒƒƒƒ ƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒ —ƒƒ ƒ

dwSSBCaps
ƒƒƒƒ ƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒ”—

dwSSBCKeyCaps
ƒƒƒƒ ƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒƒƒ ƒ ŒŠ

dwSSBCFXCaps
ƒƒƒƒ ƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒƒFXŒŠ

dwSSBRops[DD_ROP_SPACE]
ƒƒƒƒ ƒƒƒ ƒƒƒƒ ƒƒƒƒƒƒƒ“‘‚ƒƒƒ —ƒƒ ƒ

dwReserved4, **dwReserved5**, **dwReserved6**
—‚‚‚‚—‚‚‚‚‚

DDCOLORKEY

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

ƒ ƒ

IDirectDrawSurface2::GetColorKey
IDirectDrawSurface2::SetColorKey

dwColorSpaceLowValue
ƒƒ ƒ ‚‚‚—‚‚‚‚ ”ˆ‚‰Œ’

dwColorSpaceHighValue
ƒƒ ƒ ‚‚‚—‚‚‚‚ ”ˆ‚ Œ’

DDOVERLAYFX

```
typedef struct _DDOVERLAYFX{
    DWORD dwSize;
    DWORD dwAlphaEdgeBlendBitDepth;
    DWORD dwAlphaEdgeBlend;
    DWORD dwReserved;
    DWORD dwAlphaDestConstBitDepth;
union
{
        DWORD                dwAlphaDestConst;
        LPDIRECTDRAWSURFACE lpDDSAlphaDest;
};
    DWORD dwAlphaSrcConstBitDepth;
union
{
        DWORD                dwAlphaSrcConst;
        LPDIRECTDRAWSURFACE lpDDSAlphaSrc;
};
    DDCOLORKEY dckDestColorkey;
    DDCOLORKEY dckSrcColorkey;

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

IDirectDrawSurface2::UpdateOverlay

dwSize
‘‘,‘,, ,,ƒƒƒ, ‘‘,Ž,,,‘,’, ’,,,,,,,,,,

dwAlphaEdgeBlendBitDepth
ƒƒƒƒ ƒƒƒ ƒƒƒƒƒƒƒŒ ,Ž’,,,,,Ž—,,,ƒƒƒ “

dwAlphaEdgeBlend
ƒƒƒƒ ƒƒƒ ƒƒƒƒƒƒƒ,ƒƒƒƒ,,,Ž,,,’

dwReserved
—,Ž—,,,,——,,,,,

dwAlphaDestConstBitDepth
ƒƒƒƒ“‘ ,ƒƒƒƒ’ ,Ž’,,,,,Ž—,,,ƒƒƒ “

dwAlphaDestConst
ƒƒƒƒ“‘ ,ƒƒƒƒ ƒƒƒƒ,,,Ž,,,’

lpDDSAlphaDest
ƒƒƒƒ“‘ ,ƒƒƒƒ ƒƒƒƒ,,,Ž,,,ƒ ƒƒƒ,ƒƒƒƒ

dwAlphaSrcConstBitDepth
ƒƒƒƒ“‘Œ,ƒƒƒƒ’ ,Ž’,,,,Ž—,,,ƒƒƒ “

dwAlphaSrcConst
ƒƒƒƒ“‘Œ,ƒƒƒƒ ƒƒƒƒ,,,Ž,,,’

lpDDSAlphaSrc
ƒƒƒƒ“‘Œ,ƒƒƒƒ ƒƒƒƒ,,,Ž,,,ƒ ƒƒƒ,ƒƒƒƒ

dckDestColorkey
— ,,ƒƒƒƒ“‘ ƒƒ ƒ

dckSrcColorkey
— ,,ƒƒƒƒ“‘Œƒƒ ƒ

dwDDFX

ƒ ƒ ƒƒ FXƒƒƒ

DDOVERFX_ARITHSTRETCHY

,,Š‘,,,, ,Ž,‰,,‰ŽŠ‘,Ž—,,

DDOVERFX_MIRRORLEFTRIGHT

‰”“, ,

DDOVERFX_MIRRORUPDOWN

‰”“, ,

dwFlags

,,ƒƒƒ,Œ Ž,,,,,, ,,ƒƒƒ,,,,,,,,,

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 dwYUVAlphaBitMask;
};
} DDPIXELFORMAT, FAR* LPDDPIXELFORMAT;
```

の **IDirectDrawSurface2::GetPixelFormat** DirectDrawSurface のの

dwSize

‘‘,‘,, ,,ƒƒƒ, ‘‘,Ž,,,‘,’, ’,,,,,,,,,,

dwFlags

ƒƒƒƒƒ Œƒƒƒ

DDPF_ALPHA

ƒƒƒƒ ƒƒ ƒƒƒ,ƒƒƒƒ —ƒ ƒƒƒ,‹ ,,,,

DDPF_ALPHAPIXELS

ƒ ƒƒƒ,ƒƒƒƒ ƒƒƒƒ •,Ž,,ƒƒƒƒ ƒƒ ƒƒƒ,,,

DDPF_COMPRESSED

ƒ ƒƒƒ,Ž’,,,ƒƒ ƒƒƒ,ƒƒƒƒ ƒ ƒ,Ž,“,   ,,, —’,ƒ ƒˆ ,,

DDPF_FOURCC

FourCC ƒ ƒ,—Œ,,,

DDPF_PALETTEINDEXED1

DDPF_PALETTEINDEXED2

DDPF_PALETTEINDEXED4

DDPF_PALETTEINDEXED8

ƒ ƒƒƒ,,ƒƒƒ ,ƒƒƒ ,ƒƒƒ,,,,,ƒƒƒ,ƒƒƒƒƒƒ,,,,,

DDPF_PALETTEINDEXEDTO8

ƒ ƒƒƒ,,ƒƒƒ ,ƒƒƒ,,,, ,ƒƒƒ, ,,ƒƒƒ ƒƒƒƒ,ƒƒƒƒƒƒ,,,,

DDPF_RGB

ƒƒƒƒ ƒƒ ƒƒƒ ‘‘,,RGB ƒ ƒ,—Œ,,,

DDPF_RGBTOYUV

ƒ ƒƒƒ,RGB ƒ ƒ,Ž,“,, ,,, ,,, —,Š,YUV
ƒ ƒ,•Š,,   ,,,,ƒ ƒ,ƒƒ ƒƒƒ,ƒƒƒƒ ƒƒ ƒƒƒ ‘‘,Š,, DDPF_RGB
ƒƒƒ,,,ƒƒƒ,,,,,,,,,,

DDPF_YUV

ƒƒƒƒ ƒƒ ƒƒƒ ‘‘,, YUV ƒ ƒ,—Œ,,,

DDPF_ZBUFFER

Zƒƒƒƒ —ƒ ƒƒƒ,ƒƒƒƒ ƒƒ ƒƒƒ,‹ ,,,

dwFourCC

FourCC ƒ ƒ

dwRGBBitCount

ƒƒƒƒ–, RGB ƒƒƒ 4,8,16,24,,,,32

dwYUVBitCount

ƒƒƒƒ–, YUV ƒƒƒ DDBD_4 DDBD_8 DDBD_16 DDBD_24 ,,,, DDBD_32

dwZBufferBitDepth

Zƒƒƒƒ ƒƒƒ “ 8,16,24,,,,,32

dwAlphaBitDepth

ƒƒƒƒ ƒƒƒƒ ƒƒƒ “ DDBD_1 DDBD_2 DDBD_4 ,,,, DDBD_8

dwRBitMask

ƒƒƒ‚ƒƒƒ

dwYBitMask

‚ƒƒƒ‚ƒƒƒ

dwGBitMask

—ƒƒƒ‚ƒƒƒ

dwUBitMask

‚ƒƒƒ‚ƒƒƒ

dwBBitMask

ƒƒƒ‚ƒƒƒ

dwVBitMask

‚ƒƒƒ‚ƒƒƒ

dwRGBAlphaBitMask

ƒƒƒƒ ƒƒƒƒ‚ƒƒƒ

dwYUVAlphaBitMask

ƒƒƒƒ ƒƒƒƒ‚ƒƒƒ

DDSCAPS

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

DirectDrawSurface の DirectDraw
DDCAPS の

dwCaps

ƒ ƒƒƒ‚”—‚•‚ ˆ‰‚ƒƒƒ‚‚‚‚‚‚‚‚‚‚‚‚• ‚‘ ‚Ž‚

DDSCAPS_3D

‚‚‚‰ˆŒŠ ‚‚‚‚ƒƒ ƒ‚‚‚‚‚‚ ƒƒƒƒ ƒƒƒ‚‘‚‚ DDSCAPS_3DDEVICE
ƒƒƒ‚Ž‚‚‚‚‚‚

DDSCAPS_3DDEVICE

‚‚ƒ ƒƒƒ‚‚‚•Œ—‚——‰”‚‚‚‚•‚
ƒƒƒƒ ƒƒƒ‚ƒƒƒƒ‚“ ‚ƒ ƒ‚‚Š‚“‚‚‚‚ƒƒƒ ƒŠ‚ƒ ƒƒƒ‚Ž‚‚‚‚ “’‚ƒ ƒ‚‚‚•Œ‚‚‚‚
‚‚‚‚‚‚‚• ‚‚‚‚‚ ‚‚ƒƒƒ‚Ž—‚‚‚‚‚‚‚‚
‚‚‚‚ƒƒƒ‚ƒ ƒ‚‚‚‚ƒƒƒ‚‚‚‚‚‚‚ ƒ ƒƒƒ‚‚‚ƒ ƒ‚‚Š‚“‚‚‚‚‚

DDSCAPS_ALLOCONLOAD

ƒ ƒƒƒ‚**IDirect3Dtexture::Load**ƒƒƒƒ‚‚‚‚ƒ ƒ‚‚‚‚‚ ƒ ƒƒƒ‚ƒƒƒ‚Š‚“‚‚‚‚‚‚
Ž’

DDSCAPS_ALPHA

‚‚ƒ ƒƒƒ‚ƒƒƒƒ •‚Š‚‚‚‚‚‚‚•‚
ƒƒƒƒ ƒƒ ƒƒƒ‚‚‚ƒ ƒƒƒ‚‚‚ƒƒƒƒ •‚‚‚‚ƒƒƒƒ ƒ ƒ RGBA ‚‚‚‚ YUVA
‚‚‚‚ ‚ ‚‚‚ƒƒƒƒ •‚‚‚Š‚‚‚‚‚‚‚‚Œ’‚‚‚‚‚Ž“‚‚‚‚‚‚‚‚‚‚

DDSCAPS_BACKBUFFER

‚‚ƒ ƒƒƒ‚ƒ ƒƒƒ ƒƒƒƒƒƒ ‘‘‚ƒƒƒ ƒƒƒƒ‚‚‚‚‚‚•‚
ˆ”‚‚‚ƒƒƒ‚ DDSCAPS_FLIPƒƒƒ‚Ž—
‚‚‚**IDirectDraw2::CreateSurface**ƒƒƒƒ‚‚‚‚ ’‚‚‚ DDSCAPS_FRONTBU

FFERƒ ƒƒƒ‚‚’ ‚‚ƒ ƒƒƒ‚‚‚‚ƒƒƒ‚Ž‚‚‚
‘‚ƒ ƒƒƒ‚ DDSCAPS_FLIP DDSCAPS_BACKBUFFER
ƒƒƒ‚‘ ‚‚‚ DDSCAPS_FRONTBUFFER
‚Œ”‚‚ƒƒƒ ƒƒƒƒ‚‚‚”’‚‚‚ ‚‚”—
‚ **IDirectDraw2::CreateSurface**ƒƒƒƒ‚‘‚‚‚ “—Œ‚ƒƒƒ ƒƒƒƒ‚ ‚‚‚
‚‚ƒƒƒƒ‚Œ‚ ‚‚‚Œ ‚‚ƒ ƒƒƒ‚ ƒ ƒƒƒ ƒƒƒƒƒƒ ‘‘‚Œ ‚‚‚‚‚ƒƒƒƒ ƒƒƒƒ
‚‚‚‚‚‚ƒƒƒ ƒƒƒƒ ‚‚‚‚—•‚‚‚ƒƒƒƒ‚‚‚‚‚‚‚‚ ‚
IDirectDrawSurface2::AddAttachedSurface‚Ž ‚‚‚‚ DirectDraw
‚ƒƒƒƒƒƒ ‘‘‚”ˆ‚ ‚ƒ ƒƒƒ‚ƒƒ ƒ‚‚

DDSCAPS_COMPLEX

• ƒ ƒƒƒ‚‹ ‚‚‚‚‚‚‚‚•‚ • ƒ ƒƒƒ‚‚ˆ ‚ƒ ƒƒƒ‚ ‚‚‚‚‚
’‰ƒ ƒƒƒ‚ƒ ƒ ƒ ƒƒƒ‚ƒƒƒƒ‚‚‚ • ‘‘‚‚‚ƒ ƒ‚”Š‚‚‚‚‚‚‚‚”Š‚‚‚‚‚‚‚‚

DDSCAPS_FLIP

‚‚ƒ ƒƒƒ‚ƒ ƒƒƒƒƒƒƒƒƒ ‘‘‚ˆ•‚‚‚‚‚‚•‚ ‚‚”—
‚ **IDirectDraw2::CreateSurface**ƒƒƒƒ‚“‚‚‚‚‚ ‚‚‚‚ƒƒƒƒ ƒƒƒƒ‚‚‚‚‚ˆ ‚ƒƒƒ
ƒƒƒƒ‚ ‚‚‚ DirectDraw
‚ƒƒƒƒ ƒƒƒƒ‚DDSCAPS_FRONTBUFFERƒƒƒ‚‚‚ƒƒƒƒ ƒƒƒƒ‚—
‚‚‚‚ƒ ƒƒƒ‚ DDSCAPS_BACKBUFFERƒƒƒ‚ƒƒƒ‚‚
‚‚ƒƒƒ‚Ž’‚‚ **DDSURFACEDESC** ‘‘‚
dwBackBufferCountƒƒƒ‚ ‚‚‚‚‚‚ƒƒƒ‚‚‚‚‚‚‚‚‚
IDirectDraw2::CreateSurfaceƒƒƒƒ‚Ž—
‚‚ ‘ ‚ƒ ƒƒƒ‚ ‚Ž ‚DDSCAPS_COMPLEXƒƒƒ‚ƒƒƒ‚‚‚‚‚‚‚‚‚

DDSCAPS_FRONTBUFFER

‚‚ƒ ƒƒƒ‚ƒ ƒƒƒ ƒƒƒƒƒƒ ‘‘‚ƒƒƒƒ ƒƒƒƒ‚‚‚‚‚‚•‚
‚‚ƒƒƒ‚ˆ”‚ DDSCAPS_FLIP ”—
‚ƒƒƒ‚‚‚‚‚ **IDirectDraw2::CreateSurface**ƒƒƒƒ‚‚‚‚ƒƒƒ‚‚‚
IDirectDraw2::CreateSurfaceƒƒƒƒ‚Ž—‚‚ ‚“—Œƒƒƒƒ ƒƒƒƒ‚ ‚‚‚
‚‚ƒ ƒƒƒ‚ DDSCAPS_FLIPƒƒƒ‚Ž‚‚‚ ƒƒƒƒƒƒ ‘‘‚Œ ‚‚‚‚‚ **IDirectDrawSurface2::AddAttachedSurface** ‚Ž—‚‚‚‚‚‚‚‚‘‚ƒƒƒ ƒƒƒƒ‚ƒƒƒƒ‚‚‚

DDSCAPS_HWCODEC

‚‚ƒ ƒƒƒ‚ƒ ƒƒƒƒ‚‚‚‚ˆ ‰ ‚‚‚ƒƒƒ ƒ‚Ž‚‚‚‚‚‚‚‚‚‚•‚

DDSCAPS_LIVEVIDEO

‚‚ƒ ƒƒƒ‚ƒƒƒ ƒƒƒ‚Ž‚Ž‚‚‰”‚‚‚‚‚‚•‚

DDSCAPS_MIPMAP

‚‚ƒ ƒƒƒ‚‚ƒƒƒ‚ƒƒƒƒƒƒ‚‚‚‚‚‚•‚
‚‚ƒ ƒƒƒ‚ƒƒƒƒƒƒ‚Œ ‚‚‚‚‚‘‚DDSCAPS_MIPMAP
ƒ ƒƒƒ‚ƒƒƒƒ‚‚‚ ‚‚‚ ‘ ‚ƒ ƒƒƒ‚ ‚**IDirectDrawSurface2::AddAttachedSurface**ƒƒƒƒ‚Ž—‚‚‚‚‚–Ž“‚ ‚‚‚‚ ‚‚‚‚
IDirectDraw2::CreateSurfaceƒƒƒƒ‚Ž—‚‚‚‚‚ˆ–“‚ ‚‚‚ ‚‚‚‚ƒƒƒ‚ƒƒƒ‚‚‚‚
DDSCAPS_TEXTURE‚ƒƒƒ‚‚‚‚‚‚‚‚‚

DDSCAPS_MODEX

‚‚ƒ ƒƒƒ‚ 320×200‚‚‚‚ 320×240‚ƒ ƒX10 (ƒ ƒ X) ƒ ƒƒƒ‚‚‚‚‚‚•‚

DDSCAPS_OFFSCREENPLAIN

‚‚ƒ ƒƒƒ‚ƒ ƒ ƒƒ ƒƒƒƒƒ Zƒƒƒƒ ƒƒƒƒ ƒƒƒƒ ƒƒƒ ƒƒƒƒ ‚‚‚‚ƒƒƒƒ
ƒ ƒƒƒ‚‚‚‚‚‚‚ ƒƒƒ ƒŠ‚ƒ ƒƒƒ‚‚‚‚‚‚•‚ ƒƒ ƒ ƒ ƒƒƒ‚Ž•‚‚‚‚‚Ž‚‚‚

DDSCAPS_OVERLAY

‚‚ƒ ƒƒƒ‚ƒ ƒ ƒƒ‚‚‚‚‚‚•‚
Œ ‚ƒ ƒƒƒ‚ƒ ƒ ƒƒ‚‚‚‚‚‚‚‚‚‚‚‚ ‰Ž‚‚‚‚ ‚‚‚‚‚‚‚Œ‚‚

DDSCAPS_VISIBLE , ƒ ƒƒƒ,Œ ƒ ƒ ƒƒ,,,,,,,,,”’,,,,,Ž,,,,,,,

DDSCAPS_OWNDC

,,ƒ ƒƒƒ,’ŽŠ,,,,ƒƒƒƒ ƒƒƒƒƒƒ ,, ,,Š˜,Ž,,,,,,,,•,

DDSCAPS_PALETTE

,,ƒƒƒƒ ƒƒƒƒ,,,ˆˆ“,DirectDrawPalette
ƒƒƒƒƒƒ, ,,,,ƒ ƒƒƒ,ƒƒƒƒ,,,,,,•,

DDSCAPS_PRIMARYSURFACE

,,ƒ ƒƒƒ,ƒƒƒƒƒ ƒ ƒƒƒ,,,,,,•, ,,ƒ ƒƒƒ,‰Ž,,,,,,ˆ–,,

DDSCAPS_PRIMARYSURFACELEFT

,,ƒ ƒƒƒ, ––,ƒƒƒƒƒ ƒ ƒƒƒ,,,,,,•, ,,ƒ ƒƒƒ,ƒ ƒ, –,Œ,,,,,“—,•,
,,ƒ ƒƒƒ, ,,,Ž
DDSCAPS_PRIMARYSURFACEƒƒƒ,Ž,,,,ƒ ƒƒƒ, ˆ–,,,,‰––
,ƒƒƒƒƒ ƒ ƒƒƒ,‰Ž,,,

DDSCAPS_SYSTEMMEMORY

,,ƒ ƒƒƒ ƒƒƒ,ƒƒƒƒ ƒƒƒ,Š,“,,,,,,,•,

DDSCAPS_TEXTURE

,,ƒ ƒƒƒ,,,ƒƒƒƒƒ,,,—,,,,,,,,,,•, ,,, •,,,,,ƒƒƒƒƒ,,,Ž,,,,,,,,,,,,,,,,

DDSCAPS_VIDEOMEMORY

,,ƒ ƒƒƒ,ƒƒƒƒƒƒ ƒƒƒ,‘ ,,,,,•,

DDSCAPS_VISIBLE

,,ƒ ƒƒƒ,‘,, ,,,• ,‘ ,‰–,”‰,,,,,,•, ,,,ƒƒƒƒƒ ƒ ƒƒƒ,ƒƒƒ,,,

DDSCAPS_WRITEONLY

ƒ ƒƒƒ, ,,, —,,,,,,•, ƒ ƒƒƒ,,,“, ,ƒƒƒƒ,ˆ”•Œ ,, —
Š, ,,‰” ,,, ,,ƒ ƒƒƒ,,,“, ,Œ‰,“—,• ,,,,

DDSCAPS_ZBUFFER

,,ƒ ƒƒƒ,Zƒƒƒƒ,,,,,,•, Zƒƒƒƒ,•Ž,,,, •,Š,,,, ƒƒƒƒ,‰– ‹“,–––
,,,ƒƒƒ “ •,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;
```

の

IDirectDraw2::CreateSurface

dwSize

‘‘,‘,,   ,,ƒƒƒ, ‘‘,Ž,,,‘,’,   ’,,,,,,,,,,

dwFlags

ƒƒƒƒƒ Œƒƒƒ ˆ‰,ƒƒƒ,,,,,,,,,• ,‘ ,Ž,   ‘‘,,,ƒƒƒ,—Œ,,,Ž’,,

DDSD_ALL

“—,,,,,,,ƒƒƒ,—Œ,,,,,,•,

DDSD_ALPHABITDEPTH

dwAlphaBitDepth ƒƒƒ,—Œ,,,,,,•,

DDSD_BACKBUFFERCOUNT

dwBackBufferCount ƒƒƒ,—Œ,,,,,,•,

DDSD_CAPS

ddsCaps ƒƒƒ,—Œ,,,,,,•,

DDSD_CKDESTBLT

ddckCKDestBlt ƒƒƒ,—Œ,,,,,,•,

DDSD_CKDESTOVERLAY

ddckCKDestOverlay ƒƒƒ,—Œ,,,,,,•,

DDSD_CKSRCBLT

ddckCKSrcBlt ƒƒƒ,—Œ,,,,,,•,

DDSD_CKSRCOVERLAY

ddckCKSrcOverlay ƒƒƒ,—Œ,,,,,,•,

DDSD_HEIGHT

dwHeight ƒƒƒ,—Œ,,,,,,•,

DDSD_MIPMAPCOUNT

dwMipMapCount ƒƒƒ,—Œ,,,,,,•,

DDSD_PITCH

lPitch ƒƒƒ,—Œ,,,,,,•,

DDSD_PIXELFORMAT

ddpfPixelFormat ƒƒƒ,—Œ,,,,,,•,

DDSD_REFRESHRATE

dwRefreshRate ƒƒƒ,—Œ,,,,,,•,

DDSD_WIDTH

dwWidth *ƒƒƒ*‚—Œ‚‚‚‚‚•‚

DDSD_ZBUFFERBITDEPTH

dwZBufferBitDepth *ƒƒƒ*‚—Œ‚‚‚‚‚•‚

dwHeight

ƒ ƒƒƒ‚ ‚

dwWidth

“—*ƒ ƒƒƒ*‚•

lPitch

Ž‚*ƒƒƒ*ŠŽ‚‚‹— •’‚‚ *ƒƒƒ*

dwBackBufferCount

ƒƒƒ ƒƒƒƒ‚

dwMipMapCount

ƒƒƒƒƒƒƒƒƒ‚

dwZBufferBitDepth

Z*ƒƒƒƒ*‚ “

dwRefreshRate

ƒƒƒƒƒƒ‘“ *ƒƒƒƒƒƒ ƒ ƒ*‚‹ ‚‚‚ŽŽ—‚‚‚

dwAlphaBitDepth

ƒƒƒƒ ƒƒƒƒ‚ “

dwReserved

—‚‚‚‚‚

lpSurface

Š˜‚‚‚‚‚*ƒ ƒƒƒ ƒƒƒ*‚*ƒƒƒƒ*

ddckCKDestOverlay

ƒ ƒ ƒƒ ‚Ž—‚‚*ƒƒ ƒ*

ddckCKDestBlt

ƒƒƒƒ“‘ ‚Ž—‚‚*ƒƒ ƒ*

ddckCKSrcOverlay

*ƒ ƒ ƒƒ*Œ‚Ž—‚‚*ƒƒ ƒ*

ddckCKSrcBlt

ƒƒƒƒ“‘Œ‚Ž—‚‚*ƒƒ ƒ*

ddpfPixelFormat

ƒ ƒƒƒ‚*ƒƒƒƒ ƒƒ ƒƒƒ*‹

ddsCaps

DirectDraw *ƒ ƒƒƒ*”—

•’

*IDirectDraw2 IDirectDrawSurface2 IDirectDrawPalette IDirectDrawClippe*IDirectDraw2

DD_OK

—‹, Œ,Š—,,

DDERR_ALREADYINITIALIZED

ƒƒƒƒƒƒ,,,, Š‰,,,,,

DDERR_BLTFASTCANTCLIP

DirectDrawClipper
ƒƒƒƒƒƒ,**IDirectDrawSurface2::BltFast**ƒƒƒƒ,Œ, ,,“,,,“‘Œƒ ƒƒƒ,ƒƒƒƒ,,,,,

DDERR_CANNOTATTACHSURFACE

—‹,,,ƒ ƒƒƒ,,ƒ ƒƒƒ,ƒƒƒƒ,,,,

DDERR_CANNOTDETACHSURFACE

—‹,,,ƒ ƒƒƒ,,ƒ ƒƒƒ,ƒƒƒƒ,,,,

DDERR_CANTCREATEDC

Windows,,,ˆ ,,,,,ƒƒƒƒ ƒƒƒƒƒƒ DC , ,,,,,,,,

DDERR_CANTDUPLICATE

ƒƒƒƒƒ ƒ ƒƒƒ 3Dƒ ƒƒƒ ˆ–“, ,,,ƒ ƒƒƒ,• ,,,,

DDERR_CANTLOCKSURFACE

DCI ƒƒ ƒ,,,ƒƒƒƒƒ ƒ ƒƒƒ,ƒƒƒ,,,,,,,,, ,,ƒ ƒƒƒ,,ƒƒƒƒ,‹”,,,

DDERR_CANTPAGELOCK

ƒ ƒƒƒ,ƒ ƒ ƒƒƒ,Ž”,,
ƒ ƒ ƒƒƒ,ƒƒƒƒƒƒ ƒƒƒ ƒ ƒƒƒ,,,,ƒƒƒƒ ƒ,,,ƒƒƒƒƒ ƒ ƒƒƒ, ,,“ ,,,

DDERR_CANTPAGEUNLOCK

ƒ ƒƒƒ,ƒ ƒ ƒƒƒ‰ ,Ž”,,
ƒ ƒ ƒƒƒ‰ ,ƒƒƒƒƒƒ ƒƒƒ ƒ ƒƒƒ,,,,ƒƒƒƒ ƒ,,,ƒƒƒƒƒ ƒ ƒƒƒ, ,,“ ,,,

DDERR_CLIPPERISUSINGHWND

ƒƒƒƒƒ ƒƒƒƒ,,,,ƒƒƒ ,,,,DirectDrawClipper
ƒƒƒƒƒƒ,ƒƒƒƒƒƒ ƒƒƒ,ƒƒƒ,,,,,,

DDERR_COLORKEYNOTSET

“‘Œƒƒ ƒ ,Ž’,,,,,,

DDERR_CURRENTLYNOTAVAIL

Œ ƒƒ ƒ,,,,,,

DDERR_DCALREADYCREATED

ƒƒƒƒ ƒƒƒƒƒƒ(DC),,,,,,ƒ ƒƒƒ,‘,•,,,,, ˆƒ ƒƒƒ,,,,,,,,,,,Ž“,,,,,,,,,

DDERR_DIRECTDRAWALREADYCREATED

DirectDrawƒƒƒƒƒƒ,,,, ,,,,,

DDERR_EXCEPTION

—‹,,, —, ,Š,—Š,” ,,

DDERR_EXCLUSIVEMODEALREADYSET

,,,”‘“ƒ ƒ,ƒƒƒ,,,,,,,,,,,, ‹’ƒƒƒ,ƒƒƒ,,,,,,

DDERR_GENERIC

–’‹,ƒƒ ‘,,,

DDERR_HEIGHTALIGN

—,,,,‹Œ, ,,•—,,,, —,” ,,,,

DDERR_HWNDALREADYSET

DirectDraw‹’ƒƒƒ ƒƒƒƒƒ ƒƒƒƒ,,,, ’,,,,, ƒƒƒƒƒ ƒƒƒƒ,ƒƒƒƒ,ƒ ƒƒƒ,,,, ,,ƒƒƒƒ,•Ž,,,,Š,ƒƒƒƒ,,,,

DDERR_HWNDSUBCLASSED

DirectDraw‹’ƒƒƒ ƒƒƒƒƒ ƒƒƒƒ, ƒƒƒƒƒ‰,,,,,,, DirectDraw, ‘,ƒƒƒƒ,‘Ž,,,

DDERR_IMPLICITLYCREATED

ˆ–“, ,,,ƒ ƒƒƒ,ƒƒƒƒ,,,,,,

DDERR_INCOMPATIBLEPRIMARY

ƒƒƒƒƒ ƒ ƒƒƒ —‹,Š‘,ƒƒƒƒƒ ƒ ƒƒƒ,ˆ’,,,,,

DDERR_INVALIDCAPS

ƒ ƒƒƒƒŠ ,“,,,,,,ˆ ,”—ƒƒƒ,• ,,,

DDERR_INVALIDCLIPLIST

DirectDraw ,—,,,,ƒƒƒƒƒƒ ƒƒƒ,ƒƒ ƒ,,,

DDERR_INVALIDDIRECTDRAWGUID

DirectDrawCreate Š ,“,,,ƒƒ ƒƒ ƒƒ ƒŽ•Ž GUID ,—Œ, DirectDraw ƒƒƒƒŽ•Ž,,,,

DDERR_INVALIDMODE

DirectDraw ,—‹,,,ƒ ƒ,ƒƒ ƒ,,,

DDERR_INVALIDOBJECT

DirectDraw ,–Œ,DirectDraw ƒƒƒƒƒƒ,ƒƒƒƒ,Ž,Ž,,

DDERR_INVALIDPARAMS

ƒƒƒƒ,“,,,,,,ˆ ,ƒƒƒ ƒ, ,,,,

DDERR_INVALIDPIXELFORMAT

Ž’,,,ƒƒƒƒ ƒƒ ƒƒƒ,–Œ,,,

DDERR_INVALIDPOSITION

“‘ ,ƒ ƒ ƒƒ,ˆ’,• ,,,

DDERR_INVALIDRECT

—,,,,‹Œ,–Œ,,,

DDERR_INVALIDSURFACETYPE

ƒ ƒƒƒ,Šˆ,,ƒƒƒ,,,,,, —‹,,, —,Ž ,,,,,,

DDERR_LOCKEDSURFACES

,,,ˆ ,ƒ ƒƒƒ,ƒƒƒ,,,,, —‹,,, —,Ž”,,

DDERR_NO3D

,,ƒ ƒƒƒƒ,,,,ƒƒƒƒ ƒƒƒ,‘ ,,,

DDERR_NOALPHAHW

ƒƒƒƒ ƒƒƒƒƒ ƒƒƒ ƒ ƒƒƒƒ,‘ ,,,,,,,,——,,,,,,,, —‹,,, —,Ž”,,

DDERR_NOBLTHW

ƒƒƒƒ“‘ƒ ƒƒƒƒ,‘ ,,,

DDERR_NOCLIPLIST

ƒƒƒƒƒƒ ƒƒƒ,——,,,,

DDERR_NOCLIPPERATTACHED

,,ƒ ƒƒƒ ƒƒƒƒƒƒ,,DirectDrawClipper ƒƒƒƒƒƒ,ƒƒƒƒ,,,,

DDERR_NOCOLORCONVHW

•Šƒ ƒƒƒƒ,‘ ,,, ,,,,——,,,,,, —,Ž ,,,,

DDERR_NOCOLORKEY

ƒ ƒƒƒ,Œ ƒƒ ƒ ,Ž,,,,,

DDERR_NOCOLORKEYHW

“‘ ƒƒ ƒ ,‘,,ƒ ƒƒƒƒƒƒ ƒ,,,,, —,Ž ,,,,

DDERR_NOCOOPERATIVELEVELSET

IDirectDraw2::SetCooperativeLevelƒƒƒƒ,Œ, ,,,,, Š ,Œ, ,,,

DDERR_NODC

,,, ,,,,,,,ƒ ƒƒƒ—, ,,,,,,,

DDERR_NODDROPSHW

DirectDraw ƒƒƒ — ROP ƒ ƒƒƒƒ,——,,,,

DDERR_NODIRECTDRAWHW

ƒ ƒƒƒƒ — DirectDraw ƒƒƒƒƒƒ ,,,, ƒƒƒƒ,ƒ ƒƒƒƒ,ƒƒ ƒ,,,

DDERR_NODIRECTDRAWSUPPORT

Œ ,ƒƒƒƒƒƒ ƒƒƒƒ,DirectDraw,ƒƒ ƒ,,,,,

DDERR_NOEMULATION

ƒƒƒƒƒƒ ƒƒƒƒ ƒƒƒ,——,,,,

DDERR_NOEXCLUSIVEMODE

ƒƒƒƒ ƒƒƒ,‘,”“ƒ ƒ,—‹,,,, ƒƒƒƒ ƒƒƒ,”“ƒ ƒ,Ž,,,,,

DDERR_NOFLIPHW

‰Žƒ ƒƒƒ,ƒƒƒƒƒƒ,,,,

DDERR_NOGDI

GDI ,‘ ,,,

DDERR_NOHWND

ƒƒƒƒƒƒ’’,ƒƒƒƒƒ ƒƒƒƒ,•—
,,, ,,,,ƒƒƒƒƒ ƒƒƒƒ,‹’ƒƒƒ ƒƒƒƒƒ ƒƒƒƒ,,,,,,,,ƒƒƒ,,,,,,

DDERR_NOMIPMAPHW

ƒƒƒƒƒƒ ƒƒƒƒƒ ƒƒƒƒƒ,ƒ ƒƒƒƒ,‘ ,,,,,,,,,,——,,,,,, —,Ž ,,,,

DDERR_NOMIRRORHW

ƒƒ ƒƒƒ ƒ ƒƒƒƒ,‘ ,,,,,,,,——,,,,,, —,Ž ,,,,

DDERR_NOOVERLAYDEST

“‘ Š—
,,,,**IDirectDrawSurface2::UpdateOverlay**ƒƒƒƒŒ, ,,,,,,,ƒ ƒ ƒƒ,‘, **IDirectDrawSurface2::GetOverlayPosition**ƒƒƒƒ,Œ, ,,,

DDERR_NOOVERLAYHW

f f ff f ffff,‘ ,,,,,,,,,——,,,,,, —,Ž ,,,,

DDERR_NOPALETTEATTACHED

ffff ffffff,,,*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 *ffffff*, —‹,,,,*fff ffff*,Ž,,,,,

DDERR_NOT4BITCOLORINDEX

DirectDrawSurface *ffffff*, —‹,,,,*fff*,*ff ffffff ffff*,Ž,,,,,

DDERR_NOT8BITCOLOR

DirectDrawSurface *ffffff*, —‹,,,,*fff ffff*,Ž,,,,,

DDERR_NOTAOVERLAYSURFACE

”*f f ff f fff*,‘, *f f ff fff fff*,Œ, ,,,

DDERR_NOTEXTUREHW

fffff 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 ffffff f*,,,,, —,Ž ,,,,

DDERR_NOZBUFFERHW

Z*ffff*,‘,,*f ffff ff f*,,,,, *ffffff fff*,,Z*ffff* ,,,,Z*ffff*,Ž— ,,*ffff*“‘,Ž ,,,,

DDERR_NOZOVERLAYHW

ƒ ƒ ƒƒ ƒ ƒƒƒ, ƒ ƒƒƒƒ,ƒ ƒ ƒƒ Zƒ ƒ ,ƒƒ ƒ,,,, Zƒ ƒ ,–Ž,,

DDERR_OUTOFCAPS

—‹,,, —,•—,ƒ ƒƒƒƒ,,,Š,“,,,,,,

DDERR_OUTOFMEMORY

DirectDraw, —, ,,, •,ƒƒƒ,,,

DDERR_OUTOFVIDEOMEMORY

DirectDraw, —, ,,, •,ƒƒƒƒƒƒ ƒƒƒ,,,

DDERR_OVERLAYCANTCLIP

ƒ ƒƒƒƒ,ƒƒƒƒƒƒ ƒ ƒ ƒƒ,ƒƒ ƒ,,,

DDERR_OVERLAYCOLORKEYONLYONEACTIVE

ƒ ƒ ƒƒ ,,,,ˆ ,—Œ,ƒƒ ƒ ,Ž,,,,,

DDERR_OVERLAYNOTVISIBLE

IDirectDrawSurface2::GetOverlayPosition
ƒƒƒƒ,•‰Žƒ ƒ ƒƒ ,Œ, ,,,

DDERR_PALETTEBUSY

ƒƒƒƒ,,,,,,,ƒƒƒƒ,,,,ƒƒƒ,,,,,,, ƒƒƒƒ,‹”,,,

DDERR_PRIMARYSURFACEALREADYEXISTS

,, —,,,,ƒƒƒƒƒ ƒ ƒƒƒ, ,,,,

DDERR_REGIONTOOSMALL

IDirectDrawClipper::GetClipListƒƒƒƒ,“,,,—ˆ, ,,,,

DDERR_SURFACEALREADYATTACHED

,,,ƒƒƒƒ,,,,,,,,,,ƒ ƒƒƒ,ƒ ƒƒƒ,ƒƒƒƒ,,,,,,

DDERR_SURFACEALREADYDEPENDENT

,,,‘,ƒ ƒƒƒ,ˆ‘,,,,ƒ ƒƒƒ,ˆ‘,,,,,,,

DDERR_SURFACEBUSY

ƒ ƒƒƒ,,,,,,,ƒƒƒƒ,,,,ƒƒƒ,,,,,,, ƒ ƒƒƒ,,ƒƒƒƒ,‹”,,,

DDERR_SURFACEISOBSCURED

ƒ ƒƒƒ,,ƒƒƒƒ, ƒ ƒƒƒ,•,‰,,,,,,,‹”,,,

DDERR_SURFACELOST

ƒ ƒƒƒ ƒƒƒ,•‘,,, ƒ ƒƒƒ,,ƒƒƒƒ,‹”,,, ,,ƒ ƒƒƒ,•,DirectDrawSurface
ƒƒƒƒƒƒ,,**IDirectDrawSurface2::Restore**ƒƒƒƒ,Œ, ,,,,,,,,,

DDERR_SURFACENOTATTACHED

—‹,,,ƒ ƒƒƒ,ƒƒƒƒ,,,,

DDERR_TOOBIGHEIGHT

DirectDraw ,,,,—‹,,, ,,‘,,,,

DDERR_TOOBIGSIZE

DirectDraw ,,,,—‹,,,‘,,,‘,,,, ,,, Œ•, ,,•,–‘,,,

DDERR_TOOBIGWIDTH

DirectDraw ,,,,—‹,,,•,‘,,,,

DDERR_UNSUPPORTED

ƒƒ ƒ,,,,,, —,,,,

DDERR_UNSUPPORTEDFORMAT

—‹,,, FourCC ƒƒ ƒƒƒ,DirectDraw ,,,,ƒƒ ƒ,,,,,,

DDERR_UNSUPPORTEDMASK

—‹,,,ƒƒƒƒ ƒƒ ƒƒƒ,, ƒƒƒƒƒƒ ,DirectDraw ,,,,ƒƒ ƒ,,,,,,

DDERR_UNSUPPORTEDMODE

ƒƒƒƒƒƒ,Œ ƒƒ ƒ,,,,,,ƒ ƒ,,,

DDERR_VERTICALBLANKINPROGRESS

’‹ ŠŠ,,,

DDERR_WASSTILLDRAWING

,,ƒ ƒƒƒ,, ,,,ƒ ƒƒƒ,,‘‰,ƒƒƒƒ“‘‘ ,Š—,,,,,

DDERR_WRONGMODE

ˆ,,,ƒ ƒ, ,,,,, ,,ƒ ƒƒƒ,ƒƒƒƒ,,,,,,,,,

DDERR_XALIGN

—,,,,‹Œ,•—,,,,”ˆ, •ˆ—,•,,,,,,,,,,,