

Part B

Microsoft(DirectX(3
ffffffŠ"fff

,.,***ffffff***,<□,,,,,□●,□—
Š“,,,,,Ž •–,,,• “,,,,,,,,,
Microsoft , ,,ffffff,< ,,,, ffff fff,“,,,,, ’,“ · ’ Œ ‘,’“ —
Œ,Ž,,,, ,ffffff, Microsoft , –,,,‰o,,,, “< · ’ Œ ‘,’“ —Œ,‰o,—
,,,,,,,,,

(1996 Microsoft Corporation. All rights reserved.

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

,,‘, •–,‰Ž–,ŠŽ, •,,,

'5
 Direct3D•Žf f ffffff
 μŠ §
 f ffffŠ §
 IDirect3DRM"—fff ffff §
 IDirect3DRMArray §
 IDirect3DRMDeviceArray §
 IDirect3DRMFaceArray §
 IDirect3DRMFrameArray §
 IDirect3DRMLightArray §

IDirect3DRMPickedArray §

IDirect3DRMViewportArray.....	§
IDirect3DRMVisualArray.....	
IDirect3DRM.....	§
IDirect3DRMAnimation.....	§
IDirect3DRMAnimationSet.....	§
IDirect3DRMDevice.....	§
IDirect3DRMFace.....	§
IDirect3DRMFrame.....	§
IDirect3DRMLight.....	§
IDirect3DRMMaterial.....	§
IDirect3DRMMesh.....	§
IDirect3DRMMeshBuilder.....	§
IDirect3DRMObject.....	§
IDirect3DRMShadow.....	
IDirect3DRMTexture.....	§
IDirect3DRMUserVisual.....	§
IDirect3DRMViewport.....	§
IDirect3DRMWinDevice.....	§
IDirect3DRMWrap.....	§
“.....	§
—<œ.....	§
„,œ.....	§
•'.....	§

Š

xe "Direct3DRMCreate"§

Direct3DRMCreate

HRESULT Direct3DRMCreate(LPDIRECT3DRM FAR * lpD3DRM);

Direct3DRMffffff,ffffff, ,,

•.....□Œ,,,DP_OK□Ž”,,,ff ,•, •’,^—,,,,, Direct3D•Žf f,•’,Ž ,,,,

lpD3DRM

Œ, , , Œ, , —Œ,Direct3DRMffff,“,ffff,ffff

Ž Direct3DRMObject

xe "D3DRMColorGetAlpha"§

D3DRMColorGetAlpha

D3DVALUE D3DRMColorGetAlpha(D3DCOLOR d3drmc);

,ffff—‘,Ž“,,

•..... Œ,,,ffff’,•, Ž”,,,0,•,□

d3drmc

ffff—‘,Ž“,,□□

Ž□□D3DRMColorGetBlue, D3DRMColorGetGreen, D3DRMColorGetRed

xe "D3DRMColorGetBlue"§

D3DRMColorGetBlue

D3DVALUE D3DRMColorGetBlue(D3DCOLOR d3drmc);

□, —‘,Ž“,,

•..... Œ,,, ’,•, Ž”,,,0,•,

d3drmc

—‘,Ž“,,

Ž D3DRMColorGetAlpha, D3DRMColorGetGreen,

D3DRMColorGetRed

xe "D3DRMColorGetGreen"§

D3DRMColorGetGreen

D3DVALUE D3DRMColorGetGreen(D3DCOLOR d3drmc);

,——‘,Ž“,,

•..... □Œ,,,—□’,•, Ž”,,,0,•,

d3drmc

```

——‘,Ž“,,
Ž D3DRMColorGetAlpha, D3DRMColorGetBlue, D3DRMColorGetRed

```

```

xe "D3DRMColorGetRed"
D3DRMColorGetRed
D3DVALUE D3DRMColorGetRed(D3DCOLOR d3drmc);

```

```

, ——‘,Ž“,,
•..... Ą,, ,•, Ž”,,,0,•,

```

```

d3drmc
——‘,Ž“,,
Ž D3DRMColorGetAlpha, D3DRMColorGetBlue, D3DRMColorGetGreen

```

```

xe "D3DRMCreateColorRGB"§
D3DRMCreateColorRGB
D3DCOLOR D3DRMCreateColorRGB(D3DVALUE red, D3DVALUE green,
D3DVALUE blue);

```

```

—,,,□□——,, ——‘,, RGB , , ,
•..... Ą,, ,,RGB’,•, Ž”,,,0,•,

```

```

red, green, blue
RGB , ——‘
Ž D3DRMCreateColorRGBA

```

```

xe "D3DRMCreateColorRGBA"§
D3DRMCreateColorRGBA
D3DCOLOR D3DRMCreateColorRGBA(D3DVALUE red, D3DVALUE green,
D3DVALUE blue, D3DVALUE alpha);

```

```

—,,,□□—□□□□,,,ffff——‘,, RGBA , , ,
•..... Ą,, ,,RGBA’,•, Ž”,,,0,•,

```

```

red, green, blue, alpha
RGBA , ——‘
Ž□□D3DRMCreateColorRGB

```

```

xe "D3DRMFREEFUNCTION"§
D3DRMFREEFUNCTION
typedef VOID (*D3DRMFREEFUNCTION)(LPVOID lpArg);
typedef D3DRMFREEFUNCTION *LPD3DRMFREEFUNCTION;

```

fff,%o*,,,ŠŽ,ffff fff',',,
•.....•',,,

lpArg
ffff fff',f f,ffff
• Cffffff f ff,,• •, ffff fff, “Ž,fff%o*Š ,',,,,,,,□

xe "D3DRMMALLOCFUNCTION"§
D3DRMMALLOCFUNCTION
typedef LPVOID (*D3DRMMALLOCFUNCTION)(DWORD dwSize);
typedef D3DRMMALLOCFUNCTION *LPD3DRMMALLOCFUNCTION;

fff,Š,“,□,ŠŽ,ffff fff',',,
•..... Ć,,, Š,“,,,,fff,ffff,• Ž”,,,0,•,

dwSize
Š,“,,,,fff,‘,,,fff’^,Ž’,,
• Cffffff f ff,,•□□□ffff fff,□“Ž,fffŠ,“,Š□,’<,,,,,,,

xe "D3DRMMatrixFromQuaternion"§
D3DRMMatrixFromQuaternion
void D3DRMMatrixFromQuaternion (D3DRMMATRIX4D mat,
LPD3DRMQUATERNION lpquat);

%o“,Ž, —,ĆŽ,,
•.....•',,,

mat
Š , Ć,•,,, ĆŽ,,, —,“,ffff D3DRMMATRIX4DĆ,—,,
lpquat
D3DRMQUATERNION ‘“,ffff .

xe "D3DRMQuaternionFromRotation"§
D3DRMQuaternionFromRotation
LPD3DRMQUATERNION
D3DRMQuaternionFromRotation(LP3DRMQUATERNION lpquat,
LPD3DVECTOR lpv, D3DVALUE theta);

Ž’,,,Ž,’ , Ž’,,,Š“ ffff’^□,%o“, ,
•..... Ć,,, ‘1fff f,,“,,,4Ž —,ffff,• Ž”,,,0,•,

lpquat

‘ $\square \mathfrak{E} \%_0, \check{Z}, \check{Z}, D3DRMQUATERNION_N \square$ “ \square

lpv

$\%_0$ “ $\check{Z}, \check{Z}, \check{Z}, \check{Z}, \check{Z}, \check{Z}, \%_0$ “ \check{S} “ $\square ffff$ ” $\wedge \square \square$

```
xe "D3DRMQuaternionMultiply"$
D3DRMQuaternionMultiply
LPD3DRMQUATERNION D3DRMQuaternionMultiply(LPD3DRMQUATERNION lpq,
      LPD3DRMQUATERNION lpa, LPD3DRMQUATERNION lpb);
```

のの,

• $\mathfrak{E}_{,,}$, ‘ $ffff f_{,,}$ “ $_{,,}$ “ $_{,,}ffff, \bullet, \check{Z}$ ” $_{,,}$ 0, \bullet ,

lpq

$\check{Z}, \mathfrak{E} \%_0, \check{Z}, \check{Z}, D3DRMQUATERNION$ “ $_{,,}ffff$

lpa, lpb

$\check{S}, \text{ }_{,,}, D3DRMQUATERNION$ “

xe "D3DRMQuaternionSlerp"\$

```
D3DRMQuaternionSlerp
LPD3DRMQUATERNION
D3DRMQuaternionSlerp(LPD3DRMQUATERNION lpq,
      LPD3DRMQUATERNION lpa, LPD3DRMQUATERNION lpb, D3DVALUE alpha);
```

$\bullet_{,,}, ffff, ffff, ffff$ ” $\wedge \check{Z}$ ” $_{,,} \square$

• $\square Cffff \square f \square ff_{,,} \bullet \square \bullet$, $\square \square \square fffff \square fff$, $\square \check{Z}, fff \square \check{S}, \text{ }_{,,}, \check{S} \square, ' < , , , \square$

```
xe "D3DRMVectorAdd"$
```

```
D3DRMVectorAdd
```

```
LPD3DVECTOR D3DRMVectorAdd(LPD3DVECTOR lpd, LPD3DVECTOR
      lps1,
```

```
      LPD3DVECTOR lps2);
```

$_{,,}, ffff, \%_0 \check{Z},$

• $\square \mathfrak{E}_{,,}, \square '1fff \square f_{,,}$ “ $_{,,}, fff$ ” $_{,,}$ 0

lpd

$\%_0 \check{Z}, \mathfrak{E} \%_0, \check{Z}, \check{Z}, D3DVECTOR$

lps1, lps2

D3DVECTOR

xe "D3DRMVectorCrossProduct"§

D3DRMVectorCrossProduct

LPD3DVECTOR D3DRMVectorCro3DVECTOR
lps2);

,,,ffff,Š□,œŽ,,□

- □œ,,□`1fff□f,,,`\\,,,ffff,•,□ž",,,,0,•,□

lpd
Š□,œ%, ž, ž, D3DVECTOR “

lps1, lps2
Š ,œŽ,,D3DVECTOR “

xe "D3DRMVectorDotProduct"§

D3DRMVectorDotProduct

D3DVALUE D3DRMVectorDotProduct(LPD3DVECTOR lps1,
LPD3DVECTOR lps2);

○

- ○ 0

lps1, lps2

D3DVECTOR

xe "D3DRMVectorModulus"§

D3DPD3DVECTOR lpv);

~%o,ž,,,,,fff',,•,□

- □œ,,D3DVECTOR□`\\',,,• 0

lpv

LPD3DVECTOR lpaxis, D3DVALUE theta);

Ž',,,,Ž,'□,ffff,%"',,,,□
• □Œ,,,□'1fff□f,,,□,fff,•,□Ž",,,,0,•,□

lpr
□—Œ%,Ž,Ž,D3DVECTOR □“□
lpv

の D3DVECTOR

lpaxis

の D3DVECTOR

theta

の%“ Š“

xe "D3DRMVectorScale"\$

D3DRMVectorScale

LPD3DVECTOR D3DRMVectorScale(LP3DVECTOR lpd, LPD3DVECTOR
lps,
D3DVALUE factor);

の
• □Œ, 1 0

lpd

□□□□□ Ž,or
□ŽŒ□□,,'1,,,□ff□fff,□,,,□,,'2,,,2",,,,□

xe "D3DRMVectorSubtract"\$
D3DRMVectorSubtract
LPD3DVECTOR D3DRMVectorSubtract(LP3DVECTOR lpd, LPD3DVECTOR lps1,
LPD3DVECTOR lps2);

,,,,ffffŠ,ŒŽ,□,□
• 1 0

lpd

D3DVECTOR “

lps1

lps2 ,^,,,D3DVECTOR “

lps2

```

lps1,,^,D3DVECTOR  “
f  ffffŠ

```

```

xe "D3DRMDEVICEPALETTECALLBACK"$
D3DRMDEVICEPALETTECALLBACK
void (*D3DRMDEVICEPALETTECALLBACK)
(LPDIRECT3DRMDEVICE lpDirect3DRMDev, LPVOID lpArg, DWORD
dwIndex,

```

LONG red,Ž‘,ffff□fff,’<,,,□

```

•  •’,,,□

lpDirect3DRMDev
,,ffff, IDirect3DRMDevice fff□ffff□
lpArg
f□ffffŠ□,“,,,,ffff□fff’<,f□f□
dwIndex
-<,,,ffff□ffff,ffff□
red, green, blue
Ž’,,,ffff□ffff□,□□-□□-’□
ffff,f□ffffŠ□,Æ,□,□□,Æ’,,,□  の

```

```

xe "D3DRMFRAMEMOVECALLBACK"$
D3DRMFRAMEMOVECALLBACK
void (*D3DRMFRAMEMOVECALLBACK)(LPDIRECT3DRMFRAME
lpD3DRMFrame,
LPVOID lpArg, D3DVALUE delta);

```

```

ff□ の ff,ffffff,ffffff,“—
,,,,,, ff□f□f□f,•□,,,,,,f□ffffŠ
の

```

•

lpD3DRMFrame

```

□□□□□□□□ Direc□ff□f,^,•□,,,□^“,%“,,,,,—‘,,,□Š—‘,,,•□,velocity_of_component ´
delta,•,,,□,,,^•,,,—•,`“,□”^,ff□f,“—,□’,,,,,,□,,,ffff,□fffŽŠ,“—
,,,□Ž““,□f□f,“—,`“,•Š,,,□.
fff□fff,□□f□f,“—
,f□f,‰`,□□,,,,,,□,,,,,□Š□Š,,,,□Žf の
(1 f,,,affff)´1,fff,,,,,, ,, a,—<,,,‰““,,,,ffff“,,,a´
delta,“, „•“, Žff f,“— の a´ delta , ff f,“—,(v
+ (a´ delta)) ffff,•Š,,, ,, v, ,’,Æ ,Žff f,““,““,Ž,,,,,

```

„f ff
IDirect3DRMFrame::AddMoveCallback,,,IDirec
t3DRMFrame::Del,,fffff,œ,„ž,„„„,„f
ffffš,œ,„,„

```
xe "D3DRMLOADCALLBACK"$  
D3DRMLOADCALLBACK  
void (*D3DRMLOADCALLBACK) (LPDIRECT3DRMOBJECT lpObject, REFIID  
ObjectGuid,  
    LPVOID lpArg);  
  
IDirect3DRM::Loadffff“•,fffff,f ff,„,„,fffffš,ž`,fffffff,‘<,,,„  
•
```

lpObject

Direct3DRMObject

ObjectGuid

f,ff ff ff fž•ž GUID

lpArg

f ffffš „,,,ffff fff’<,f f
ffff,f ffffš ,œ, , ,œ’,,,, ,š‘,fff fff,,,ffffff,œ , ž, ,,
„,f ffffš ,œ, ,
ž IDirect3DRM::Load

```
xe "D3DRMLOADTEXTURECALLBACK"$  
D3DRMLOADTEXTURECALLBACK  
HRESULT (*D3DRMLOADTEXTURECALLBACK) (char *tex_name,  
void *lpArg,  
    LPDIRECT3DRMTEXTURE *lpD3DRMTex);  
  
Loadffff,“•, fffff ffff ,,,,ff f ,ž „, , , ,f ffffš ,ž‘,fff  
f fff,‘<,,,
```

• „œ,,,D3DRM_OK„ž”,,,ff„,•’,^—
,,,,„Direct3D•žf„f,•’,ž„,,,„

tex_name

fffff—
lpArg

```

fffff[]ffffE-,f[]f[]
lpD3DRMTex
Direct3DRMTextureffffff[]
fffff[]ffff,[]Windowsffffff[] .bmp [] ,f[]ffff[]fffff[] .ppm [] P6
EŽ^Š,fffff,ff[]f,ffffffffff,,[],f[]ffffŠ[],Ž-,,,,,,,[]
fffff,[]f[]ffffŠ[],E,[],[],E',,,,[],Š`,ffff[]ffff,,ffffffffff,E[],
,[]Ž,[][],,,[]"

```

```

IDirect3DRM::Load, IDirect3DRMAnimationSet::Load,
IDirect3DRMFrame::Load, IDirect3DRMMeshBuilder::Load

```

```
xe "D3DRMOBJECTCALLBACK"§
```

```
D3DRMOBJECTCALLBACK
```

```

void (*D3DRMOBJECTCALLBACK)(LPDIRECT3DRMOBJECT
lpD3DRMObj,
LPVOID lpArg);

```

```

IDirect3DRM::EnumerateObjects          0
ff,—<,, ,f ffffŠ ,Ž',ffff fff'<,,

```

```

ffffffffff,—,IDirt3DRMObjectfff[]ffff[]ffff[]fff,[]
—<,,Šffffff,,,Releaseffff,E,[],,,,,,,[]
lpArg
f[]ffffŠ[],`,,,,fffff[]fff'<,f[]f[]
fffff,f[]ffffŠ[],E,[],[],E',,,,[],Š`,ffff[]ffff,,ffffffffff,E[],[]Ž,[][],,,[]",f[]f
fffŠ[],E,[],[]
Ž[]IDirect3DRM::EnumerateObjects

```

```

xe "D3DRMUPDATECALLBACK"§
D3DRMUPDATECALLBACK
void (*D3DRMUPDATECALLBACK) (LPDIRECT3DRMDEVICE lpobj,
LPVOID lpArg,
int iRectCount, LPD3DRECT d3dRectUpdate);

```

```

ffff,• ,,,,ffff fff,E , ,f ffffŠ ,Ž',ffff fff'<,,
• •',,,

```

```
lpobj
```

```

f ffffŠ ,“—,Direct3DRMDeviceffffff
lpArg
f ffffŠ ,“,,,ffff fff'<,f f
iRectCount
d3dRectUpdate          <E,
d3dRectUpdate
,,,—^,• , ,^ ,D3DRECT “,”— •,ffff^,Ž',,,

```

ffff, f fff
,, ,Š',fff fff,,fffff,Œ , Ž, , , ,f ffffŠ ,Œ, ,
Ž□□ID:DeleteUpdateCallback, IDirect3DRMDevice::Update

xe "D3DRMUSERVISUALCALLBACK"\$
D3DRMUSERVISUALCALLBACK
int (*D3DRMUSERVISUALCALLBACK) (LPDIRECT3DRMUSERVISUAL lpD3DRMUV,
LPVOID lpArg, D3DRMUSERVISUALREASON lpD3DRMUVreason,
LPDIRECT3DRMDEVICE lpD3DRMDev, LPDIRECT3DRMVIEWPORT lpD3DRMview);

f□f∅∅<,,

•

lpD3DRMUVreasonfff f,D3DRMUSERVISUAL_CANSEE,Ž',, f f f
ffff ffffff,ff f f“,‰Ž ‘,,, TRUE,•, ,,,,,,FALSE,•,
lpD3DRMUVreasonfff f,D3DRMUSERVISUAL_RENDER,
fff'<,',,, , , ,TRUE,•,,,^,,,

lpD3DRMUV
Direct3DRMUserVisual
lpArg

lpD3DRMUVreason

D3DRMUSERVISUALREASON □□□∅∅□□□ □□
,□f, ff□f□f“,‰Ž□',,,,□ffff□fff,TRUE,•,□, ,□□ffff□fff,□lpD3DRMview
fff□f,Ž' , , ,ffff,Ž-, , □

D3DRMUSERVISUAL_RENDER

fffff□fff,□f□f□fffff□fffff,fffff,,,,,,,,,□,□□ffff□fff,□lpD3DRMDevff
f□f,Ž' , , ,ffff,Ž,□

lpD3DRMDev

Direct3DRMUserVisual ff,fffff,,,,,Ž,,,Direct3DRMDevicefffff

lpD3DRMview

Direct3DRMUserVisual ffffff,‰Ž ‘,,,,Œ,,,,,Ž,,, Direct3DRMViewport
fffff

ffff,f□ffffŠ□,Œ,□,□□,
' ,fff□fff,,fffff,Œ□,□Ž,□□□,,,□”,f□ffffŠ□,Œ,□,
IDirect3DRMUserVisual::Init

xe "D3DRMWRAPCALLBACK"\$

D3DRMtor,

```
int* lpU, int* lpV, LPD3DVECTOR lpD3DRMVA, LPD3DVECTOR lpD3DRMVB,
LPVOID lpArg);
```

```
,,fffffŠ,f,f,,,,,
IDirect3DRM"-fffff
"-fffff,-,,fffff,fffff,"-ff%,,Š',ff`',-,"-,,,,,,^%
,"-fffff,ž-,,,,,,
IDirect3DRMArray
```

IDirect3DRMDeviceArray

IDirect3DRMFaceArray

IDirect3DRMFrameArray

IDirect3DRMLightArray

IDirect3DRMPickedArray

IDirect3DRMViewportArray

IDirect3DRMVisualArray

IDirect3DRMArray

```
IDirect3DRMArrayffffffffff,f
      □□                □□□□
fff,'ž,,□,,ffffffff,"□,,-fffff,—
,,□,,ffffrrayffffffff,GetSizeffff,ff□f,,□
,,,COMfffffff,"-
□IDirect3DRMArrayfffff,IUnknownffffffffff,®,,,□,,fffff,ž,,,
ffff,f,f,f,,□
AddRef
```

QueryInterface

Release

xe "IDirect3DRMArray\:\.GetSize"§

IDirect3DRMArray::GetSize

DWORD GetSize();

Direct3DRMArray fffffff,fff, fffffff'^,Ž“,,,

- f

IDirect3DRMDeviceArray

ffff□ffffff,f□f%□,□,□IDirect3DRMDeviceArrayfff□ffff□ffff,ffffff,□Š”□,□,□Š—
,□□IDirect3DRMDevice,IDirect3DRMDeviceArray fff□ffff□,ž□,,,□
IDirect3DRMDeviceArray fff□ffff,□ž,ff

GetElement

GetSize

IDirect3DRMDeviceArray の COM IUnknown
f ffff,Œ ,, IUnknownfff ffff,ž,,,ffff,ff f,,
AddRef

QueryInterface

Release

lpD3DRMDevice
IDirect3DRMDevice fff ffff,,fff

DRMDeviceArray::GetSize

DWORD GetSize();

Direct3DRMDeviceArray fffffff,Š,,-‘□,Ž“,,,□
• —‘□,*,□

IDirect3DRMFaceArray

IDirect3DRMFaceArrayfff□ffff,ffff,——
,,□,,ffff,□IDirect3DRMFaceArrayfff□ffff□ffff,fffff,,□Š”,,,,Š—
,□□IDirect3DRMFace,IDirect3DRMFaceArrayfff□ffff □,Ž□,,,
IDirect3DRMFaceArrayfff ffff, Ž,ffff,ff f,,

GetElement

GetSize

IDirectfff□ffff□ffff,Œ□,,□IUnknownfff□ffff,□Ž,,,,ffff,ff□f,,□
AddRef

QueryInterface

Release

,,,□IDirect3DRMFaceArrayfff□ffff,□IDirect3DRMObjectfff□ffff,,Ž,ffff,Œ
□,,□
AddDestroyCallback

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMFaceArrayfffff, IDirect3DRMMeshBuilder::GetFacesfff,Œ, ,
,,,Ž“,,,

xe "IDirect3DRMFaceArray\:\:GetElement"\$

IDirect3DRMFaceArray::GetElement

HRESULT GetElement(DWORD index, LPDIRECT3DRMFACE *
lpD3DRMFace);

Direct3DRMFaceArray Ž“,,

• □Œ,,,D3DRM_OK□Ž”,,,ff□,•,□•’,^—
,,,,,DRMFace

IDirect3DRMFacefff□ffff,,ffff□

xe "IDirect3DRMFaceArray\:\:GetSize"\$

IDirect3DRMFaceArray::GetSize

DWORD GetSize();

Direct3DRMFaceArrayfffff,š,,,–\□,Ž“,,,□

• —\□,•,□

IDirect3DRMFrameArray

ff f fffffff,ff f%o IDirect3DRMFrameArray の
,,ffffff, IDirect3DRMFrameArrayfff ffff ffff,ffffff,, Š”,,,,Š
—, IDirect3DRMFrame,IDirect3DRMFrameArrayfff ffff ,Ž”,,,,
IDirect3DRMFrameArrayfff fff fff f f,,
GetElement

GetSize

IDirect3DRMFrameArray ffff, ,,,COMfff ffff,“—
IUnknownfff ffff ffff,Œ ,, IUnknown
,,,ffff,ff f,,
AddRef

QueryInterface

Release

□
,,□IIDirect3DRMFrameArrayfff□fddDestroyC
allback

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMFrameArrayfffff, IDirect3DRMPickedArray::GetPick,,, IDirect3DRMFrame::GetChildren

xe "IDirect3DRMFrameArray\::GetElement"§

IDirect3DRMFrameArray::GetElement

HRESULT GetElement(DWORD index, LPDIRECT3DRMFRAME *
lplpD3DRMFrame);

Direct3DRMFrameArrayfffff,Ž',,,—
• CE,,,D3DRM_OK •, •',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,,

index

lplpD3DRMFrame

IDirect3DRMFrame fff ffff,,ffff,Ž,Ž,ffff

Direct3DRMFrameArrayfffff,Š,,,,—'□,Ž",□
• —'□,•,□

IDirect3DRMLightArray

CEfffff,ff f%,,,, IDirect3DRMLightArrayfff ffff,ffff,—
,, ,ffff, IDirect3DRMLightArrayfff ffff ffff,ffffff,, Š" の
IDirect3DRMLight IDirect3DRMLightArrayfff ffff ,Ž ,,,

IDirect3DRMLightArrayfff ffff, Ž,ffff,ff f,,

GetElement

GetSize

IDirect3DRMLightArrayfff□ffff,□,,,COMfff□ffff,
 IUnknownfff□ffff□ffff,Œ□,,□IUnknownfff□ffff,Ž,,,fff

AddRef

QueryInterface

Release

GetName

SetName

Direct3DRMLightArray fffffff,ž',,-,ž",,□
 • D3DRM_OK
 Direct3D の

index

lpD3DRMLight
IDirect3DRMLight fff ffff,,fff,Ž,Ž,fff

xe "IDirect3DRMLightArray\:\:GetSize"\$
IDirect3DRMLightArray::GetSize
DWORD GetSize();

Direct3DRMLightArray fffff,Š,,,,,—‘ ,Ž“,,,
• —‘ ,•,

IDirect3DRMPickedArray
fff□fffff,ff□ IDirect3DRMPickedArrayfff□ffff,fff,—
,,□,,ffff,□IDirect3DRMPickedArray
f,,□Š”,,,,,Š—,□□IDirect3DRMPickedArrayfff□ffff □,Ž□
IDirect3DRMPickedArray の
GetPick

GetSize

IDirect3DRMPickedArray の COM □□□□□□□□□□IUnknown □□□□□□□□□□□□□□
□□IUnknown □□□□□□□□
,Ž,,□IDirect3DRMPickedArrayfff□ffff,□IDirect3DRMObjectfff□ffff,,Ž,ffff,☒, ,□
AddDestroyCallback

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMPickedArrayffffff, IDirect3DRMViewport::Pick

xe "IDirect3DRMPickedArray\:\:GetPick"§

IDirect3DRMPickedArray::GetPick

HRESULT GetPick(DWORD index, LPDIRECT3DRMVISUAL * lpVisual,
LPDIRECT3DRMFRAMEARRAY * lpFrameArray,
LPD3DRMPICKDESC lpD3DRMPickDesc);

Ž',,,,fff,,,CE ,Direct3DRMVisual,Direct3DRMFrameffffff,Ž“,,,
• ,D3DRM_OK Ž”,,,,ff ,•, •',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,,

index

Ž•,, fff”—,ffffff

lpVisual

Ž',,,,fff,Š~•,,,,Direct3DRMVisual ④ ffff,ffff

lpFrameArray

Ž',,,,fff,Š~•,,,,Direct3DRMFrameArrayffffff,ffff,ffff

lpD3DRMPickDesc

fff,^~ – Ž“,,ffffff,ff fŽ•Ž,Ž',,D3DRMPICKΔESC “,,fff

Ž IDirect3DRMViewport::Pick

xe "IDirect3DRMPickedArray\:\:GetSize"§

IDirect3DRMPickedArray::GetSize

DWORD GetSize();

Direct3DRMPickedArray fffffff,Š,,,,,—‘ ,Ž“,,,

•

IDirect3DRMViewportArray

IDirect3DRMViewportArray
IDirect3DRMViewportArray の
IDirect3DRMViewportArray の
IDirect3DRMViewportArray

IDirect3DRMViewportArray
GetElement

GetSize

IDirect3DRMViewportArray
IUnknown
COM
IUnknown
AddRef

QueryInterface

Release

IDirect3DRMViewportArray
IDirect3DRMObject の
AddDestroyCallback

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMViewportArray

IDirect3DRM::CreateFrame

xe "IDirect3DRMViewportArray\.\.GetElement"\$§

IDirect3DRMViewportArray::GetElement

HRESULT GetElement(DWORD index, LPDIRECT3DRMVIEWPORT * lpD3DRMViewpor, □

- □Ⓔ,,,D3DRM_OK□Ž",,,,ff□,•,□•',^-,,,,,,□□Direct3D•Žf□f,•'□,Ž□,,,,□

index

"—'\□

lpD3DRMViewpor

IDirect3DRMViewpor fff□ffff,Ž,Ž,ffff,,ffff□

xe "IDirect3DRMViewporArray\:\:GetSize"\$

IDirect**3DRMViewporArray::GetSize**

DWORD GetSize());

Direct3DRMViewporArray

-

IDirect3DRMVisualArray

fffff fffffff,ff f%

,,,, IDirect3DRMVisualArrayfff ffff,ffff,—,の

IDirect3DRMVisualArrayfff ffff ffff,ffffff,,, Š",,,,,Š—,

IDirect3DRMVisual IDirect3DRMVisualArrayfff ffff ,Ž ,,,,

GetElement

GetSize

IDirect3DRMVisualArrayfff□ffff,□,,,COMfff□ffff,“—

IUnknownfff□ffff□ffff,Ⓔ□,,□IUnknown のの

AddRef

QueryInterface

Release

<p> Clone e </p>	<p> DeleteDestroyCallback </p>
------------------------------------------	---------------------------------------

IDirect3DRMVisualArray
 AddDestroyCallback

IDirect3DRMObject の

```

Direct3DRMVisualArrayfffff, IDirect3DRMFrame::GetVisualsfffff,E,□,,,,,Ž",,,,□
xe "IDirect3DRMVisualArray\:\:GetElement"$
IDirect3DRMVisualArray::GetElement
HRESULT GetElement(DWORD index, LPDIRECT3DRMVISUAL *
lpD3DRMVisual);
Direct3DRMVisualArrayfffff
• D3DRM_OK
Direct3D の
index
lpD3DRMVisual
IDirect3DRMVisual fff ffff,Ž,Ž,ffff
  
```

xe "IDirect3DRMVisualArray\:\:GetSize"§

IDirect3DRMVisualArray::GetSize

DWORD GetSize();

Direct3DRMVisualArray ffffff,Š,,,,—‘ ,Ž“,,

•

IDirect3DRM

IDirect3DRM fff, □ Direct3DRM ffffff, □ □, □ ffff □ f
の

ff, □ IDirect3DRM fff □ ffff, ffff, ffffff,, □ Š”,,,,, Š—, □ IDirect3DRM

IDirect3DRM の

CreateAnimation

CreateAnimationSet

CreateDevice

CreateDeviceFromClipper

CreateDeviceFromD3D

CreateDeviceFromSurface

GetDevices

EnumerateObjects

CreateFace

CreateFrame

CreateLight

CreateLightRGB

CreateMaterial

CreateMesh

CreateMeshBuilder

の

CreateObject

CreateUserVisual

GetNamedObject

Load

Tick

AddSearchPath

GetSearchPath

SetSearchPath

CreateShadow

Cree

LoadTextureFromResource

SetDefaultTextureColors

SetDefaultTextureShades

fff
CreateViewport

fff
CreateWrap

IDirect3DRMfff, , , , COMfff, “–
IUnknownfff, , IUnk**nown** の

AddRef

QueryInterface

Release

IDirect3DRM COM

Direct3DRMCreate の

```
xe "IDirect3DRM\:\AddSearchPath"§
```

```
IDirect3DRM::AddSearchPath
```

```
HRESULT AddSearchPath(LPCSTR lpPath);
```

```
CE□,ffffCE□ff,□CE,ffffff,ff
• CE,,,D3DRM_OK Ž”,,,ff ,•, •,^—
,,,, Direct3D•Žf f,•’ ,Ž ,,,,
```

```
lpPath
```

```
CE ,CE ff,%o,,ff,Ž’,, NULL, ,,•Ž—,,ffff
```

```
Windows,,□ff,ffff□;□,□,□,fffff,fff,,,,,,,,□
IDirect3DRM::SetSearchPath
```

```
xe "IDirect3DRM\:\CreateAnimation"§
```

```
IDirect3DRM::CreateAnimation
```

```
HRESULT CreateAnimation(LPDIRECT3DRMANIMATION *
lpD3DRMAnimation);
```

の Direct3DRMAnimation

• □□□□□ **D3DRM_OK**のDirect3D の
□□□□□,,tion **fff**□**ffff**,Ž,Ž,**ffff**,,,**ffff**□

```
xe "IDirect3DRM\:\CreateAnimationSet"§
```

```
IDirect3DRM::CreateAnimationSet
```

```
HRESULT CreateAnimationSet (LPDIRECT3DRMANIMATIONSET *
lpD3DRMAnimationSet);
```

```
<,Direct3DRMAnimationSetffffff,□□,,□
```

```
• □CE,,,D3DRM_OK□Ž”,, , Direct3D
```

```
lpD3DRMAnimationSet
```

```
IDirect3DRMAnimationSet
```

```
fff,Ž,Ž,ffff,,ffff
```

```
xe "IDirect3DRM\:\CreateDevice"§
```

```
IDirect3DRM::CreateDevice
```

HRESULT CreateDevice(DWORD dwWidth, DWORD dwHeight,
LPDIRECT3DRMDEVICE* lplpD3DRMDevice);

Windows ,,,,,,

xe "IDirect3DRM\:\CreateDeviceFromClipper"§

IDirect3DRM::CreateDeviceFromClipper

HRESULT CreateDeviceFromClipper(LPDIRECTDRAWCLIPPER
lpDDClipper,
LPGUID lpGUID, int width, int height,
LPDIRECT3DRMDEVICE * lplpD3DRMDevice);

Ž',,,DirectDrawClipper Direct3DRM Windows

- D3DRM_OK
Direct3D の

lpDDClipper

DirectDrawClipper

lpGUID

GUIDのの NULL

width , height

lplpD3DRMDevice

IDirect3DRMDevice

の

lpGUID NULL ,

D3DPRIMCAPS

D3DPCAPCAPS_LESSEQUAL

D3DPMISCCAPS_CULLCCW

D3DPRASSTERCAPS_FOGVERTEX

D3DPSHADECAPS_ALP

HAFLATSTIPPLED

D3DPTADDRESSCAPS_WRAP

D3DPTBLENDCAPS_COPY | D3DPTBLENDCAPS_MODULATE

D3DPTTEXTURECAPS_PERSPECTIVE |
D3DPTTEXTURECAPS_TRANSPARENCY
D3DPTFILTERCAPS_NEAREST

*f ffff ffff,Œ,,,,,, ffff fff ,ffffff ffff,f f,, ,ffffff ',fff,ff
f,,“•,—,Ž,ffff fff, lpGUID,NULL,Ž’,‘,,,ffff,—,,,,,,*

xe "IDirect3DRM\:\:CreateDeviceFromD3D"\$

IDirect3DRM::CreateDeviceFromD3D

HRESULT CreateDeviceFromD3D(LPDIRECT3D lpD3D,

LPDIRECT3DDEVICE lpD3DDev, LPDIRECT3DRMDEVICE *
lpD3DRMDevice);

Ž’,,,Direct3Dffffff,Ž,, Direct3DRM Windowsffff,

,

- D3DRM_OK
Direct3D の

lpD3D

Direct3D

lpD3DDev

Direct3D

lpD3DRMDevice

IDirect3DRMDevice の

**xe "IDirect3DRM\:\:CrFromSurface(LPGUID
lpGUID, LPDIRECTDRAW lpDD,**

LPDIRECTDRAWSURFACE lpDDSBack,
LPDIRECT3DRMDEVICE * lpD3DRMDevice);

Ž',,, DirectDrawf□fff,,□ffffff,~,Windowsffff,□□,,□
• □Œ,,D3DRM_OK□Ž"„ff□,•,□'ˆ—,,,,□□Direct3D•Žf□f,•'□,Ž□,,,,□

lpGUID

Ž',,,ffff□ffff,ff□ff□ GUID

NULL

lpDD

DirectDraw の DirectDraw

lpDDSBack

fff ffff,•,DirectDrawSurfaceffffff

lpD3DRMDevice

Œ, ,, Œ,,,,,IDirect3DRMDevicefff ffff,Ž,Ž,ffff,,ffff

xe "IDirect3DRM\:\:CreateFace"§

IDirect3DRM::CreateFace

HRESULT CreateFace(LPDIRECT3DRMFACE * lpD3drmFace);

IDirect3DRMFace

• Œ,, D3DRM_OK
Direct3D•Žf f,•' ,Ž ,,,,

lpD3drmFace

Œ, ,, Œ,,,,,IDirect3DRMFacefff ffff,Ž,Ž,ffff,,ffff

```
xe "IDirect3DRFrame,
    LPDIRECT3DRFRAME*
    lpD3DRFrame);
```

```
Ž',,,□ff□f,□,,Žff□f,□□,,□
```

```
• □E,,,D3DRM_OK□Ž",,,,ff□,•,□•',^—,,,,,□□Direct3D•Žf□f,•'□,Ž□,,,,□
```

```
lpD3DRFrame
```

```
□,,ff□f,□ff□f□
```

```
lpD3DRFrame
```

```
E,□,,□E,,,,IDirect3DRFramefff□fff,Ž,Ž,ffff,,fff□
```

```
Žff□f,□f□f,"□`□,□E□,,□,,□f□f,Ž',,,,`",,,,,□□□Žff□f,
```

```
"",`",",□,,□□f□f,%",,,,,,□Žff□f,□f□f,E`,`
```

```
NULL
```

```
IDirect3DRFrame::AddChildfff,—,,,ff f, ff f,Š~*,,,,,,
```

```
Ž IDirect3DRFrame::AddChild
```

```
xe "IDirect3DRM\:\CreateLight"$
```

```
IDirect3DRM::CreateLight
```

```
HRESULT CreateLight(D3DRMLIGHTTYPE d3drmltLightType,
```

```
D3DCOLOR cColor, LPDIRECT3DRMLIGHT* lpD3DRMLight);
```

```
Ž',,,,fff, ,,,E□, ,,
```

```
• □E,,,D3DRM_OK □Ž",,,,ff ,•, •',^—  
,,,,, Direct3D•Žf f,•',Ž ,,,,
```

```
d3drmltLightType
```

```
D3DRMLIGHTTYPE—(□E,—,,,□E□,fff,,,
```

```
cColor
```

```
□E,
```

```
lpD3DRMLight
```

```
□E, , □E,,, IDirect3DRMLight fff,Ž,Ž,ffff,,fff
```

```
xe "IDirect3DRM\:\CreateLightRGB"$
```

```
IDirect3DRM::CreateLightRGB
```

```
HRESULT CreateLightRGB(D3DRMLIGHTTYPE ltLightType, D3DVALUE  
vRed,
```

**D3DVALUE vGreen, D3DVALUE vBlue,
LPDIRECT3DRMLIGHT* lpD3DVALUE**

xe "IDirect3DRM\:\CreateMesh"§

IDirect3DRM::CreateMesh

HRESULT CreateMesh(LPDIRECT3DRMMESH* lpD3DRMMesh);

-,ž,,,□,,,ffff□fffff,□□,,,□ffff,ff□f,%,,,,,,•ž,,,□

- □Œ,,,D3DRM_OK□ž",,,,ff□,•,□•',^—,,, Direct3D の

lpD3DRMMesh

Œ,,,,,IDirect3DRMMeshfff ffff,ž,ž,ffff,,ffff

xe "IDirect3DRM\:\CreateMeshBuilder"§

IDirect3DRM::CreateMeshBuilder

HRESULT CreateMeshBuilder(LPDIRECT3DRMMESHBUILDER*
lpD3DRMMeshBuilder);

□,,ffff□

- Œ,,,D3DRM_OK
Direct3D•žf f,•',ž ,,,,

lpD3DRMMeshBuilder

Œ, ,, Œ,,,,,IDirect3DRMMeshBuilderfff ffff,,ffff,ž,ž,ffff

xe "IDirect3DRM\:\CreateObject"§

IDirect3DRM::CreateObject

HRESULT CreateObject(REFCLSID rclsid, LPUNKNOWN pUnkOuter,
REFIID riid, LPVOID FAR* ppv);

- D3DRM_OK
Direct3D の

rclsid

の

pUnkOuter

COM

riid

 \mathcal{O}

ppv

```
□の□□□□□□□□□□□□□□□□□□□□□□□□  
□□□□□□□□□□□□□□□□□□IDirect3DRM の  
□□□□□□□□□□□□□□□□□□□□□□□□□□ □  
□□□□□□□□□□□□□□□□Init □□□□□□□□□□  
Init □□□□□□□□□□□□□□ ffxe  
"IDirect3DRM:\:CreateShadow"$
```

IDirect3D8::CreateShadow

```
HRESULT CreateShadow(LPDIRECT3DRMVISUAL lpVisual,
    LPDIRECT3DRMLIGHT lpLight, D3DVALUE px, D3DVALUE py, D3DVALUE pz,
    D3DVALUE nx, D3DVALUE ny, D3DVALUE nz,
    LPDIRECT3DRMVISUAL * lplpShadow);
```

ž', , , f f f f f ,

,,,fffff,,,

- $\text{E}_{\text{D3DRM_OK}} \rightarrow \text{Direct3D}$

lpVisual

```
%%,€, Direct3DRMVisual ffffff
```

lpLight

```

CEE,,,IDirect3DRMLight fff  ffff

```

 p_x, p_y, p_z nx, ny, nz

00,,•-,-

lplpShadow

Œ, , Œ,,, %%%f f f f f,—Œ,f f f f, Š%%,,f f f f,,f f f f

$$xe \text{ "IDirect3D7M:\:\:CreateTexture"}\S$$

IDirect3D7::CreateTexture

HRESULT CreateTexture(LPD3DRMIMAGE lpImage,

```
LPDIRECT3DTEXTURE* lpD3DRMTexture);
```

**fff,ff,f,,ff•',^—
,,,,, Direct3D•Žf,f,•',Ž,,,,,**

lpImage

```
ffff,f,f,•,D3DRMIMAGE``  
lpD3DRMTexture  
E,, ,E,,,, IDirect3DRMTexturefff,fff,Ž,Ž,fff,,fff  
ffff,fffff,,,, Direct3DRM,fff,ff, ,fff,,ff,f,Š~•,,,,fff,—  
,,, ,,,ff,f,fffff,ffff,fff,Ž—,,,,,  
  
xe "IDirect3DRM\\:CreateTextureFromSurface"$  
IDirect3DRM::CreateTextureFromSurface  
  
HRESULT CreateTextureFromSurface(LPDIRECTDRAWSURFACE lpDDS,  
    LPDIRECT3DRMTEXTURE * lpD3DRMTexture);
```

Ž',,, DirectDraw fff,ffff, ,,
• E,,,D3DRM_OKŽ",,,ff,•,•',^—,,,,, Direct3D•Žf,f,•',Ž,

```
lpDDS  
    DirectDrawSurfacefffff  
    lpD3DRMTexture  
E,, ,E,,,, IDirect3DRMTextureff  
  
xe "IDirect3DRM\\:CreateUserVisual"$  
IDirect3DRM::CreateUserVisual  
    HRESULT CreateUserVisual(D3DRMUSERVISUALCALLBACK fn,
```

**LPVOID lpArg,
LPDIRECT3DRMUSERVISUAL *
l fff',ffff,—,fffff,,,**

```
• E,,,D3DRM_OKŽ",,,ff,•,•',^—,,,,, Direct3D•Žf,f,•',Ž,,,,  
  
fn  
fffff fff'<, D3DRMUSERVISUALCALLBACKfffffŠ  
lpArg  
ffffffŠ, ",,fffff fff'<, f f  
lpD3DRMUV  
E, , E,,,, IDirect3DRMUserVisual f
```

```
xe "IDirect3DRM\\:CreateViewport"$  
IDirect3DRM::CreateViewport  
HRESULT CreateViewport(LPDIRECT3DRMDEVICE lpDev,
```

```
LPDIRECT3DRMFRAME lpCamera, DWORD dwXPos,
DWORD dwYPos, DWORD dwWidth, DWORD dwHeight,
LPDIRECT3DRMVIEWPORT* lplpD3DRMViewport);
```

```
ffff, (dwXPos, dwYPos), (dwXPos + dwWidth, dwYPos + dwHeight)
,ff f f, , , ^,ffff ,•,•,•,
• D3DRM_OK Ž",,,ff ,•, •',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,
```

```
lpDev
ff f f, ,,,ffff
lpCamera
```

ff□,^',•Œ,<□,,ff□f,,ffff□lpD3DRMViewport

```
Œ,□,,□Œ,,,,IDirect3DRMViewportfff□ffff,ž,ž,ffff,,ffff□
fff□f□f,□fff,š,f□f□,fffffff,"' , ,□f□,•Œ,□ffff,fff,,ž", , ,□
```

```
xe "IDirect3DRM\\:\\:CreateWrap"$
```

```
IDirect3DRM::CreateWrap
```

```
HRESULT CreateWrap(D3DRMWRAPTYPE type, LPDIRECT3DRMFRAME
lpRef,
```

```
D3DVALUE ox, D3DVALUE oy, D3DVALUE oz, D3DVALUE dx,
D3DVALUE dy,
```

```
D3DVALUE dz, D3DVALUE ux, D3DVALUE uy, D3DVALUE uz,
D3DVALUE ou,
```

```
D3DVALUE ov, D3DVALUE su, D3DVALUE sv,
```

```
LPDIRECT3DRMWRA* lplpD3DRMWrap);
```

```
□□□□,,,,ž—
,,fff<" ,□□,,□fff,Œ“,ffff [ox oy oz] zž, [dx
dy dz] □yž, [ux uy uz] ,—,,,□2Dffff [ou ov]
, [su sv]•M_OK□Ž",,,ff□,•,□•',^—
,,,,□□Direct3D•Žf□f,•'□,ž□,,,,□
```

type

```
D3DRMWRAPTYPE-<Œfff, , , ,□
```

```
lpRef
```

```
fff, , , ,ž□ff□f, ,ffff□
```

```
ox, oy, oz
```

```
fff,Œ"□
```

```
dx, dy, dz
```

fff,zŽ□
ux, uy, uz
fff,yŽ□
ou, ov
fffff,Œ"□
su, sv
fffff,□ŽŒ□
lpD3DRMWrap

IIDirect3DRMWrap fff,Ž,Ž,ffff,,ffff
Ž IIDirect3DRMWrap

xe "IIDirect3DRM\:\EnumerateObjects"§
IIDirect3DRM::EnumerateObjects
HRESULT EnumerateObjects(D3DRMOBJECTCALLBACK func, LPVOID
 lpArg);

fffff,Direct3DRMfffff,, func

• Œ,,,D3DRM_OK Ž",,,ff ,•, •',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,

func

ŠDirect3DRMObjectffff□□
lpArg
f□ffffŠ□,“,,,□ffff□fff'◁,f□f□

xe "IIDirect3DRM\:\GetDevices"§
IIDirect3DRM::GetDevices
HRESULT GetDevices(LPDIRECT3DRMDEVICEARRAY* lpDevArray);

ffff,□□,,,,,, Direct3DRM ffff,•,□
• Œ,,,D3DRM_OK□Ž",, Direct3D

lpDevArray
Direct3DRMffff,"—
 ,Ž,Ž,ffff Direct3DRMDeviceArrayffffff,,,,, •, IIDirect3DRMDevic
 eArrayfff ffff ,Ž ,,,

xe "IIDirect3DRM\:\GetNamedObject"§
IIDirect3DRM::GetNamedObject

lpvObjID

d3drmLOFlags
D3DRMLOAD_BYPOSITION
,,fff f, ffff,'',ffffff, ,Ž,DWORD',,ffff,, ,fff f,,NULL,Ž',
,,,,,

lpGUIDs

f f,,fff ffffŽ•Ž,—
,,ffff ,,, ,fff f,IID_IDirect3DRMMeshBuilder,IID_IDirect3DRMAni
mationSet,Š,—,, ,,,fff fff fff,ff
,

dwcGUIDs

lpGUIDs fff f,Ž',,—,—‘

d3drmLOFlags

f f fffff, D3DRMLOADOPTIONSŒ,’

d3drmLoadProc

Ž',,ffffff,ffff,“, ,,,Œ,□,,,□D3DRMLOADCALLBACK

lpArgLP

D3DRMLOADCALLBACK f□ffffŠ

d3drmLoadTextureProc

D3DRMLOADTEXTURECALLBACK ,,, “,ff fff,•—
,,ffffff,—,,fffff,“, ,,,Œ, ,,, ,fff f,,NULL,Ž',,,,,,,

lpArgLTP

D3DRMLOAD□,,fff□f,□•,□Direct3DRMAnimationSet,Direct3DRMFra
meffffff,□□, , , , , □, NULL, Ž', , , , □, , fff□f, □, , , , ffffff,f
f,, ,—,, ,fff f,,NULL,Ž',,,,,,,

xe "IDirect3DRM\:\LoadTexture"§

IDirect LPDIRECT3DRMTEXTURE*
lpD3DRMTexture);

Ž',,ffff,,fffff,f□f,,□,,fffff,□ffff“, ,8□24□,,,32ffff,fff□,Ž,,,,,,□,,□fff
ff,Windowsffffff□.bmp□,f□fff□ffffff□.ppm□P6ŒŽ,,,,,,□
• □Œ,,,D3DRM_OK□Ž“, , , f Direct3D

lpFileName

.bmp .ppm の

lpD3DRMTexture

Direct3DRMTexture の

```

xe "IDirect3DRM\:\LoadTextureFromResource"$
IDirect3DRM::LoadTextureFromResource
HRESULT LoadTextureFromResource(HRSRC rs,
    LPDIRECT3DRMTEXTURE * lpD3DRMTexture);

Ž',,,ff f,,fffff,f f,,
• Ɛ,,,D3DRM_OK Ž",,,ff ,•, •',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,,

rs
ff f,ffff
lpD3DRMTexture
Direct3DRMTextureffff,,ffff

xe "IDirect3DRM\:\SetDefaultTextureColors"$
IDirect3DRM::SetDefaultTextureColors
HRESULT SetDefaultTextureColors(DWORD dwColors);

```

Direct3DRMTexture•',^—
,,,,,□□ Direct3D•Žf□f,•'□,Ž□,,,,□

dwColors

```

□,□□
,,ffff,□IDirect3DRM::CreateTextureffff,\,Ɛ,□,,,,,,fffff,□,"%
,,,□,,,□□,,,fffff,\,,,Ɛ,,,,□

xe "IDirect3DRM\:\SetDefaultTextureShades"$
IDirect3DRM::SetDefaultTextureShades
HRESULT SetDefaultTextureShades(DWORD dwShades);

```

Direct3DRMTexture Ž—,,fffff,%o%o, ',,

- Ɛ,,,D3DRM_OK Ž",,,ff ,•, •',^—
 ,,,, Direct3D•Žf f,•' ,Ž ,,,,

dwShades
 %o%o,

```
xe "IDirect3DRM\:\.SetSearchPath"§
IDirect3DRM::SetSearchPath
HRESULT SetSearchPath(LPCSTR lpPath);
```

pPath

AddPositionKey

AddRotateKey

AddScaleKey

DeleteKey

の

SetFrame

SetTime

GetOptions

SetOptions

IDirect3DRMAnimation *COM*
IUnknown

IUnknown
のの

AddRef

QueryInterface

Release

IDirect3DRMAnimation	IDirect3DRMObject
fff,,Ž,ffff,Œ ,,	Clone
AddDestroyCallback	

GetAppData	DeleteDestroyCallback
-------------------	------------------------------

GetClassName

GetName	SetName
----------------	----------------

SetAppData

Direct3DRMAnimationfffff, IDirect3DRM::CreateAnimationfff,Œ, ,,,,Ž“,,,

**xe "IDirect3DRMAnimation\:
AddPositionKey"**

IDirect3DRMAnimation::AddPositionKey

HRESULT AddPositionKey(D3DVALUE rvTime, D3DVALUE rvX,
D3DVALUE rvY, D3DVALUE rvZ);

- D3DRM_OK Dire
- ct3DŽf f,Ž ,,,

rvTime

fff fff,Žf , ,,,,ŽŠ 0,Œ
“,,,Ž,ŽŠ,Ž, ,,,

,, の rvTime 99 rvTime 49
rvX, rvY, rvZ

3D

IDirect3DRM

Animation::DeleteKey

xe "IDirect3DRMAnimation\
\\AddRotateKey"§

IDirect3DRMAnimation::AddRotateKey

HRESULT
AddRotateKey(D3DVALUE
rvTime, D3DRMQUATERNION
*rqQuat);

fff fff,%“f ,’%,,

•

Œ,,,D3DRM_OK Ž”,,,ff ,
•, •’,

^— Direct3D の

rvTime

の 0 の の の の rvTime 99
rvTime 49

rqQon::DeleteKey

xe "IDirect3DRMAnimation\:\:AddScaleKey"\$

IDirect3DRMAnimation::AddScaleKey

HRESULT AddScaleKey(D3DVALUE rvTime, D3DVALUE rvX, D3DVALUE rvY,
D3DVALUE rvZ);

fff fff, ff f f ,<%o

• Ć,,D3DRM_OK Ź", , の Direct3D
の, Ź ,,,

rvTime

fff fff, ff f f ,<%o
,,,,ŹŠ 0,Ć",,,",^,ŹŠ,Ź',,, Ć,f ,rvTime,'99,,, rvTime,49,,,f ,ff
f fff,,,,,Ź,,

rvX, rvY, rvZ

ŹĆ .
,,ffff,,, Ź•Š,—" ,,, •Š,Š, —ĆŹ,,,,, 3D•Š ,Ź ,,,
Ź IDirect3DRMAnimation::DeleteKey

xe "IDirect3DRMAnimation\:\:DeleteKey"\$

IDirect3DRMAnimation::DeleteKey

HRESULT DeleteKey(D3DVALUE rvTime);

f ,fff fff,, ,,

• Ć,,D3DRM_OK Ź",,,ff ,•, •',^—, Direct3D
の

rvTime

```
xe "IDirect3DRMAnimation\:\:GetOptions"$
IDirect3DRMAnimation::GetOptions
D3DRMANIMATIONOPTIONS GetOptions();
```

Ž □ □ IDirect3DRMAnimatn::SetOptions xe "IDirect3DRMAnimation\:\:SetFrame"\$

```
IDirect3DRMAnimation::SetFrame
HRESULT SetFrame(LPDIRECT3DRMFRAME lpD3DRMFrame);
```

```
fff□fff,ff□f,□',,,□
•   Ē,,,D3DRM_OK                               Direct3D
    f,•' ,Ž ,,,,
```

```
lpD3DRMFrame
fff fff, ',ff f,Ž',• ,ffff
```

```
xe "IDirect3DRMAnimation\:\:SetOptions"$
IDirect3DRMAnimation::SetOptions
HRESULT SetOptions(D3DRMANIMATIONOPTIONS d3drmanimFlags);
```

```
fff□fff
•   Ē,,,D3DRM_OK Ž",,,ff ,•, •',^—
    ,,,, Direct3D•Žf f,•' ,Ž ,,,,
```

```
d3drmanimFlags
fff fff fffffŽ',,D3DRMANIMATIONOPTIONSĒ,',ffff
```

```
Ž   IDirect3DRMAnimation::GetOptionst
xe "IDirect3DRMAnimation\:\:SetTime"$
IDirect3DRMAnimation::SetTime
HRESULT SetTime(D3DVALUE rvTime);
```

の

- D3DRM_OKのDirect3D の

rvTi,f□,fff□fff,,,,,'Š,,,□

IDirect3DAnimationSet

ffff□fff,□Direct3DAnimationffffff,ff□f%
,,,,,IDirect3DAnimationSetfff□ffff,ffff,Ž–
,,,,,,□,,fff□ffff,,,,•Ž,fff□fff,—
,,□□,,,,,Š',,,□,,fffff,□IDirect3DAnimationSetfff□f
fff□ffff,ffffff,,□Š", ④ IDirect3DAnimation
IDirect3DAnimationSetfff ffff ,Ž ,,,
IDirect3DAnimationSetfff ffff,ffff, ^%o,ff f,•—,,,,,,
'‰ f f
AddAnimation

DeleteAnimation

Load

ŽŠ

SetTime

IDirect3DAnimationSetfff ffff, ,,,COMfff
IUnknownfff ffff ffff,Ĉ ,, IUnknownfff ffff,Ž,,,,ffff,ff f,,

AddRef

QueryInterface

Release

```

,, IDirect3DRMAnimationSetfff ffff, IDirect3DRMObjectfff ffff,
,Z,fff,Œ ,,

```

```

AddDestroyCallback

```

```

Clone

```

```

DeleteDestroyCallback

```

```

GetAppData

```

```

GetClassName

```

```

GetName

```

```

SetAppData

```

```

SetName

```

```

Direct3DRMAnimationSet

```

```

IDirect3DRM::CreateAnimationSet

```

```

xe "IDirect3DRMAnimationSet\:\:AddAnimation"§

```

IDirect3DRMAnimation);

fff fff,fff fff fff,%o,,
• D3DRM_OKŽ",,,ff,•,•',^-,,,,Direct3DŽff,•',Ž,.,.,

lpD3DRMAnimation
Direct3DRMAnimationfffff

xe "IDirect3DRMAnimationSet\:\:DeleteAnimation"\$
IDirect3DRMAnimationSet::DeleteAnimation
HRESULT DeleteAnimation(LPDIRECT3DRMANIMATION
lpD3DRMAnimation);

fff fff,fff fff f
•

D3DRM_OKŽ",,,ff ffect3DRMAnimatio
nfffff

xe "IDirect3DRMAnimationSet\:\:Load"\$
IDirect3DRMAnimationSet::Load
HRESULT Load(LPVOID lpvObjSource, LPVOID lpvObjID,
D3DRMLOADOPTIONS d3drmLOFlags,
D3DRMLOADTEXTURECALLBACK d3drmLoadTextureProc, LPVOID
lpArgLTP,
LPDIRECT3DRMFRAME lpParentFrame);

fff fff fff,f f,,
• D3DRM_OKのDirect3D の

lpvObjSou,fff f,Ž',,
lpvObjID
f f,ffffff,-'^',fff f,fff f,d3drmLOFlagsfff f,Ž,Ž•Žfff,,
,,Ž,.,.,D3DRMLOAD_BYPOSITIONfff,Ž',,.,fff f,ffff,'

, f f f f f f, □□, Ž, DWORD' , , f f f f

NULL

d3drmLOFlags

f f f f f f, Ž, D3DRMLOADOPTIONSŒ, '

d3drmLoadTextureProc

D3DRMLOADTEXTURECALLBACKf f f f Š , , , „, ff f f f, •—
 , , f f f f f f, —, f f f f f, „, , , , Œ, , , , , f f f f, , NULL, Ž' , , , , , , ,

lpArgLTP

D3DRMLOADTEXTURECALLBACK

○ f

lpParentFrame

□ f f □ f, f f f f □, , f f f □ f, □ f f f □ f f f f f f, f f □ f, □ □ f
 f □ f, N, f □ f, , □

xe "IDirect3DRMAnimationSet\:\:SetTime"§

IDirect3DRMAnimationSet::SetTime

HRESULT SetTime(D3DVALUE rvTime);

f f f □ f f f, Œ□, Ž Š, □' , , □

- Œ, , , D3DRM_OK
 ○' , Ž , , ,

Direct3D

rvTime

, Ž Š

IDirect3DRMDevice

f f f f f f f, — f f f f, 'Œ' , , , , IDirect3DRMDevicef f f f f f f f f f, Ž—
 , , , f f f f f f, IDirect3DRMDevicef f f f f f f f f f f f f f f f f, , Š' , , , , Š—
 , IDirect3DRMDevice , IDirect3DRMDeviceArrayf f f f f f f f f f
 , Ž , , , ,

IDirect3DRMDevicef f f f f f f f f f, ^%, ff f, •—, , , , , , , ,

f f f f f f f f f f

GetBufferCount

SetBufferCount

f

GetColorModel

GetDither

SetDither

Init

InTrianglesDrawn

GetViewports

GetWidth

GetWireframeOptions

Update

''

AddUpdateCallback

DeleteUpdateCallback

ff の

GetQuality

SetQuality

GetShades

SetShades

fffff, *Ž*

GetTextureQuality

SetTextureQuality

IIDirect3DRMDevicefff ffff, ,,,COMfff ffff,“—
IUnknownfff ffff ffff,Œ ,, IUnknownfff ffff,Ž,,,ffff,ff f,
,

AddRef

QueryInterface

Release

,,, IIDirect3DRMDevicefff ffff, IIDirect3DRMObjectfff ffff,,Ž,fff
f,Œ ,,
AddDestroyCallback

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMDevice\fffff, □ IDirect3DRM::CreateDevice

xe "IDirect3DRMDevice\:\AddUpdateCallback"§

IDirect3DRMDevice::AddUpdateCallback

**HRESULT AddUpdateCallback(D3DRMUPDATECALLBACK
d3drmUpdateProc, LPVOID arg);**

IDirect3DRMDevice::Update

- **D3DRM_OK**
Direct3D の

d3drmUpdateProc

の D3DRMUPDATECALLBACK

arg

の

IDirect3DRMDevice::DeleteUpdateCallback,
IDirect3DRMDevice::Update, D3DRMUPDATECALLBACK

xe "IDirect3DRMDevice\:\DeleteUpdateCallback"§

IDirect3DRMDevice::DeleteUpdateCallback

HRESULT

DeleteUpdateCallback(D3DRM
UPDATECALLBACK
d3drmUpdateProc,
LPVOID **arg**);

IDirect3DRMDevice::AddUpdate
Callback

ff

- D3DRM_OK
Direct3D *の*

d3drmUpdateProc

D3DRMUPDATECALLBACK *の*

arg

ID

irect3DRMDevice::AddUpdate
Callback,
IDirect3DRMDevice::Update,
D3DRMUPDATECALL

BACK

xe "IDirect3DRMDevice\::GetBufferCount"\$

IDirect3DRMDevice::GetBufferCount
DWORD GetBufferCount();

IDirect3DRMDevice::SetBufferCount

,Ž“,,

-

ffff, ,*, 1,ffff ffff 2,2 fff
f ^%03 4 ,,,

```
xe "IDirect3DRMDevice\:  
\\GetColorModel"$  
IDirect3DRMDevice::GetColorMo  
del  
D3DCOLORMODEL  
GetColorModel();
```

```
ffff,ff  fff,Ž“,,
```

•

Direct3D

RGB

D3DCOLORMODEL

```
xe "IDirect3DRMDevice\:\\GetDirect3DDevice"$
```

```
IDirect3DRMDevice::GetDirect3DDevice
```

```
HRESULT Ge の
```

- Ą,,D3DRM_OK Ž”,,,ff ,• , •’,^—
,,,, Direct3D•Žf f,•’ ,Ž ,,,,

```
lpD3DDevice
```

```
' f f,ffff fffffff,,ffff, Š%o,,,ffff,ffff
```

```
xe "IDirect3DRMDevice\:  
\\GetDither"$...ect3DRMDevice\:  
\\GetShades"$
```

```
IDirect3DRMDevice::GetShades  
DWORD GetShades();
```

```
‰‰,□,Ž“,,□,□,□Ž—,,,,□□,•Ą,,,□
```

- の

IDirect3DRMDevice::SetShades

```
xe "IDirect3DRMDevice\\:\\:GetTextureQuality"$  
IDirect3DRMDevice::GetTextureQuality  
D3DRMTEXTUREQUALITY GetTextureQuality();
```

∞∞
∞

RGB

• **D3DRMTEXTUREQUALITY**
\\:GetViewports"\$

IDirect3DRMDevice::GetViewports

```
HRESULT GetViewports(LPDIRECT3DRMVIEWPORTARRAY* lpViewports);
```

```
ffff, ff f f, Direct3DRMViewportArray
```

- D3DRM_OK Ž", ff ,•, •, ^—
,,,, Direct3D Ž f,• ,Ž ,,,

lpViewports

```
CE, □,, □CE,,,, □—CE, Direct3DRMViewportArray ff
```

xe "IDirect3DRMDevice\\:\\:GetWidth"\$

IDirect3D •,•, □

```
xe "IDirect3DRMDevice\\:\\:GetWireframeOptions"$
```

```
IDirect3DRMDevice::GetWireframeOptions
```

```
DWORD GetWireframeOptions();
```

,ffff,fff ff f fffff,Ž“,,,

- ^%,',fffOR,•,,

D3DRMWIREFRAME_CULL

∞

D3DRMWIREFRAME HIDDENLINE

[illegible]

HRESULT Init(ULONG width, ULONG height);

Windows „Ž“,,,,,,

```
xe "IDirect3DRMDevice\::InitFromClipper"$S
IDirect3DRMDevice::InitFromClipper
HRESULT InitFromClipper(LPDIRECTDRAWCLIPPER lpDDClipper,
    LPGUID lpGUID, int width, int height);
```

DirectDrawClipperffffff,,ffff, Š%,,

- E,,D3DRM_OK Ž”,,,ff ,•, •,^—
,,,,, Direct3D•Žf f,•, Ž ,,,,

lpDDClipper

□ Š%o

DirectDrawClipper

lpGUID

[illegible]

❏❏ ect3DRMDevice::InitFromD3D

```
HRESULT InitFromD3D(LPDIRECT3D lpD3D, LPDIRECT3DDEVICE
lpD3DIMDev);
```

Direct3D

- $\mathbb{E}_{,,D3DRM_OK} \check{Z}^{,,,ff} \cdot, \cdot, ^\wedge -$
 $,,,, \text{Direct3D} \cdot \check{Z} f f, \cdot, \check{Z} ,,,,$

```
lpD3D
•Žf f ffff, Š‰,,,,Ž,Direct3' f f fffffff,,ffff
lpD3DIMDev
```

•Žf f ffff, Š%,,,,Ž,' f f ffff

xe "IDirerCount(DWORD dwCount);

ffff□fff,,,,œ□Ž—,,,,ffff,□,□',,□

• □œ,,,D3DRM_OK□Ž",,,,ff□,•,□•',^—
,,,,,□□Direct3D•Žf□f,•'□,Ž

dwCount

の1 2 2 3 4
1,, ffff ffff ffff, —, ,
2 ffff,3 ffff,Ž—,,ffff fff, ffff,Ž—',ffff, ',,,,,,ffff,Ž—
,,,,,,, ,,,,ffff, Šff f,,,fff,, ,,,fffff, ,œŽ,,,,,
Ž IDirect3DRMDevice::GetBufferCount
xe "IDirect3DRMDevice\:\SetDither"§
IDirect3DRMDevice::SetDither
HRESULT SetDither(BOOL bDither);

ffff,fff fff, ',,

• □œ,,,D3DRM_OK□Ž",,,,ff□,•,□•',^—,,,, Direct3D の

bDither

の TRUE

IDirect3DRMDevice::GetDither

xe "IDirect3DRMDevice\:\SetQuality"§

IDirect3DRMDevice::SetQuality

HRESULT SetQuality (D3DRMRENDERQUALITY rqQuality);

の

• □□□□□ **D3DRM_OK**□□□□□□□□□□□□□□□□
□□□□□□□□

ffffff•Ž,□ffff,ffffff□f□fff□,Ž□,,,ffffff,•Ž,□'',,,,□šffff,
"Ž,•Ž,Ž,,,□ffff,—œ,□□•Ž,,,□ffff,—œ,•Ž,",,,,□^

```

,,,,ffff,~,•Ž,’,,, “,•Ž,—,,,,ffff, ,,, ,•Ž,,,,,
Ž IDirect3DRenderDevice::GetQuality
xe "IDirect3DRenderDevice\\:\\SetShades"§
IDirect3DRenderDevice::SetShades
HRESULT SetShades(DWORD ulShades);

```

- $\square \mathbb{E}_{\dots, D3DRM} \text{ OK } \square \check{Z}''_{\dots, ff} \square$, **Direct3D** の

ulShades

```

□□□□□の□□□□□のt3DRMDevice\:  

\::SetTextureQuality"§  

IDirect3DRMDevice::SetTextureQuality  

HRESULT SetTextureQuality(D3DRMTEXTUREQUALITY tqTextureQuality);

```

ffff,fffff•ž,□',,□
• D3DRM_OK ž",,,ff ,•, •',^—
,,,, Direct3D•žf f,• ,ž ,,,,

```

    tqTextureQuality
    D3DRMTEXTUREQUALITY—(E,fff,,, fffff,D3DRMTEXTURE,,,
    Ž    IDirect3DRMDevice::GetTextureQuality
    xe "IDirect3DRMDevice\:\:Update"§
    IDirect3DRMDevice::Update
    HRESULT Update());

```

• fffffff,ffffff,,ff f,• ,, ,, ffff ffff,f ff fŠ ,’¿,,
 □€,,D3DRM_OK□Ž”,, Direct3D
 ⑦

```
D3DRMUPDATECALLBACK
    IDirect3DRMDevice::AddUpdateCallback
        IDirect3DRMDevice::AddUpdateCallback,
D3DRMUPDATECALLBACK
IDirect3DRMFace
```

IDirect3DRMFace の IDirect3DRMFace の IDirect3DRMFace
IDirect3DRMFaceArray

IDirect3DorRGB

fffff
GetMaterial

SetMaterial

fffff
GetTexture

GetTextureCoordinateIndex

GetTextureCoordinates

GetTextureTopology

SetTexture

SetTextureCoordinates

SetTextureTopology

”“,—

AddVertex

AddVertexAndNormalIndexed

GetNormal

GetVertex

GetVertexCount

GetVertexIndex

GetVertices

IDirect3DRMFacefff ffff, ,,,COMfff ffff,—
 IUnknownfff ffff ffff,Œ , IUnknown
 ,,,ffff,ff f,,
 AddRef

QueryInterface

Release

,,,IDirect3DRMFacefffIDirect3DRMObject の
AddDestroyCallback

Clone

Direct3DRMFacefffff, IDirect3DRM::CreateFaceffff,œ, ,,,,Ž“,,,

```
xe "IDirect3DRMFace\:\:AddVertex"$
IDirect3DRMFace::AddVertex
HRESULT AddVertex(D3DVALUE x, D3DVALUE y, D3DVALUE z);
```

Direct3DRMFaceff

- œ,,,D3DRM_OK Ž”,,,ff ,•, •’,^—
,,,, Direct3D•Žf f,•’ ,Ž ,,,,

x, y, z

„,“^”,x y z •

```
xe "IDirect3DRMFace\:\:AddVertexAndNormalIndexed"$
IDirect3DRMFace::AddVertexAndNormalIndexed
HRESULT AddVertexAndNormalIndexed(DWORD vertex, DWORD
normal);
```

”“,- ,Direct3DRMFaceffffff,%o,, ,,, ffff□fff,Š,,”“,ffffff,-
□,ffffff,Ž—,,□-□”“ - Direct3DRMMeshBuilder 1

• □ ex, normal

%o,,”“,-□,ffffff□

```
xe "IDirect3DRMFace\\:GetColor"$
IDirect3DRMFace::GetColor
D3DCOLOR GetColor();
```

Direct3DRMFace の

•

```
IDirect3DRMFace::SetColor
xe "IDirect3DRMFace\\:GetMaterial"$
IDirect3DRMFace::GetMaterial
HRESULT GetMaterial(LPDIRECT3DRMMATERIAL* lpMaterial);
```

Direct3DRMFaceffffff,fffff,Ž“,,

• D3DRM_OK
Direct3D の

lpMaterial

Direct3DRMMaterial の

```
IDirect3DRMFace::SetMaterial
xe "IDirect3DRMFace\\:GetNormal"$
IDirect3DRMFace::GetNormal
HRESULT GetNormal(D3DVECTOR *lpNormal);
```

Direct3DRMFace

. D3DRM_OKの
Direct3D f, ' ,Ž ,,,, ,

lpNormal

D3DVECTOR

xe "IDirect3DRMFace\:\:GetTexture"§

IDirect3DRMFace::GetTexture

HRESULT GetTexture(LPDIRECT3DRMTEXTURE* lpTexture);

Direct3DRMFace

- D3DRM_OK
Direct3D の

lpTexture

の

IDirect3DRMFace::SetTexture

xe "IDirect3DRMFace\:\:GetTextureCoordinateIndex"§

IDirect3DRMFace::GetTextureCoordinateIndex

int Get

TextureCoordinateIndex(DWORD
dwIndex);

—
,ffff,,,'“,ffffff,ffff •,Ž“,,
,,ffffff, dwIndexfff f,Ž',
,,ffffff,'%o,,
• fffff,*,

dwIndex

—,,,'“,ffffff

xe "IDirect3DRMFace\:


```

\::GetTextureCoordinates"§
IDirect3DRMFace::GetTexture
Coordinates
HRESULT
GetTextureCoordinates(DWOR
D index, D3DVALUE *lpU,
D3DVALUE *lpV);

```

```

Direct3DRMFaceffffff',",fff
ff •,Ž",,

```

•

```

œ,,D3DRM_OK Ž",,,ff •,
•',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,,

```

```

index
",ffffff
lpU, lpV
",fffff •,Ž,Ž,•

```

```

xe "IDirect3DRMFace\::
\::GetTextureTopology"§
IDirect3DRMFace::GetTexture
Topology
HRESULT
GetTextureTopology(BOOL
*lpU, BOOL *lpV);

```

```

Direct3DRMFaceffffff,fffff^
',Ž",,

```

•

D3DRM_OK

Direct3D の

lpU, lpV

IDirect3DRMFace::SetTextureTo
pology

xe "IDirect3DRMFace\
\\GetVertex"§

IDirect3DRMFace::GetVertex
HRESULT GetVertex(DWORD
index, D3DVECTOR *lpPosition,
D3DVEC

TOR *lpNormal);

Direct3DRMFace

- D3DRM_OK
Direct3D

f,• ,Ž ,,,

index

”“,ffffff

lpPosition, lpNormal

”“,^,- ,Ž,Ž,D3DVECTOR “

xe "IDirect3DRMFace\
\\GetVertexCount"§

IDirect3DRMFace::GetVertexCou
nt

```
int GetVertexCount();
```

```
Direct3DRMFace ffffff, '“, ,Ž“,,
```

- '“,

```
,
```

```
xe "IDirect3DRMFace\.\:GetVertexIndex"§
```

```
IDirect3DRMFace::GetVertexIndex
```

```
int GetVertexIndex (DWORD dwIndex);
```

```
□□□□□□□□□□□□ f f..Color
```

```
HRESULT SetColor(D3DCOLOR color);
```

```
Direct3DRMFace ffffff, □, □',,,□
```

- □E,,,D3DRM_OK□Ž",,,,ff□,•,□•',^~,,,,,□□Direct3D•Žf□f,•'□,Ž□,,,,□

```
color
```

```
□',,,□□
```

```
Ž□□IDirect3DRMFace::GetColor
```

```
xe "IDirect3DRMFace\.\:SetColorRGB"§
```

```
IDirect3DRMFace::SetColorRGB
```

```
HRESULT SetColorRGB(D3DVALUE red, D3DVALUE green, D3DVALUE  
blue);
```

```
Direct3DRMFace の R, G, B
```

- D3DRM_OK
Direct3D•Žf f,•' ,Ž ,,,,

red, green, blue
— , —‘

xe "IDirect3DRMFace\:\:SetMaterial"\$
IDirect3DRMFace::SetMaterial
HRESULT SetMaterial(LPDIRECT3DRMMATERIAL lpD3DRMMaterial);

Direct3DRMFaceffffff,ff
• ☒☒,,D3DRM_OK☒Ž”,,,ff☒,ff☒
Ž☒☒IDirect3DRMFace::GetMaterial

xe "IDirect3DRMFace\:\:SetTexture"\$
IDirect3DRMFace::SetTexture
HRESULT SetTexture(LPDIRECT3DRMTEXTURE lpD3DRMTexture);

Direct3DRMFaceffffff,ffffff,fff,,
• ☒☒,,D3DRM_OK☒Ž” の Direct3D
の

lpD3DRMTextureCoordinates"\$
IDirect3DRMFace::SetTextureCoordinates

HRESULT SetTextureCoordinates(DWORD vertex, D3DVALUE u, D3DVALUE v);

Direct3DRMFaceffffff',",ffffff☒•,☒',,☒
• D3DRM_OK
Direct3D•Žf f,•' ,Ž ,,,,

vertex
' ,",ffffff ,,,, -,ŽŠ☒,, "0 1 2,,,
u, v
Ž', ,",Š, ",ffffff •

xe "IDirect3DRMFace\:\:SetTextureTopology"\$
IDirect3DRMFace::SetTextureTopology
HRESULT SetTextureTopology(BOOL cylU, BOOL cylV);

Direct3DRMFaceffffff,ffffff^', ',,
• ☒☒,,D3DRM_OK☒Ž”,,,ff☒•,☒•',^— Direct3D の

cylU, cylV

```

...
ffff□fff,□ff□f□ffffff,ff□f,Ž□□,Œ,,,,Ž,,,,,IDirect3DRMFramefff□f
fff,ffff,Ž—
,,□,,fffff,□IDirect3DRMFramefff□ffff□ffff,ffffff,,□Š”
,,,,Š—,□ IDirect3DRMFrame      IDirect3DRMFrameArray

```

IDirect3DRMFramefff ffff ffff, ^%,ff f,•—,,,,,,,,,

”Œ

GetSceneBackground

GetSceneBackgroundDepth

SetSceneBackground

SetSceneBackgroundDepth

SetSceneBackgroundImage

SetSceneBackgroundRGB

GetColor

SetColor

SetColorRGB

fff
GetSceneFogColor

GetSr

SetSceneFogEnable

SetSceneFogMode

SetSceneFogParams

§'
AddChild

DeleteChild

GetChildren

GetParent

GetScene

☒
AddLight

DeleteLight

GetLights

Load

GetMaterialMode

SetMaterialMode

~, ~"

AddMoveCallback

AddRotation

AddScale

AddTranslation

DeleteMoveCallback

GetOrientation

GetPosition

GetRotation

Get*Velocity*

LookAt

Move

SetOrientation

Set*Position*

SetRotation

SetVelocity

•—

GetSortMode

GetZbufferMode

SetSortMode

SetZbufferMode

GetTexture

GetTexturm

Transform

ffffff□*ffffff*
AddVisual

DeleteVisual

GetVisuals

IDirect3DRMFrame*ffffff*□*ffff*,□,,,COM*ffff*□*ffff*,`-
□IUnknown*ffff*□*ffff*□*ffff*,Ⓔ□,□IUnknown*ffff*□*ffff*,ž,,,ffff,ff□f,,□
AddRef

QueryInterface

Release

	IDirect3DRMFrame	IDirect3DRMObject
∅		
AddDestroyCallback		
Clone		
DeleteDestroyCallback		
GetAppData		
GetClassName		
GetName		
SetAppData		
SetName		

rvThet.. D3DRMCOMBINETYPE—<œ,fff

rvX, rvY, rvZ

x y z•œ, Žœ, 'ç,,

Ž',,,•Š, , IDirect3DRMFramefff ffff,,,, Ž•,,, fff, •Š—,

3D

xe "IDirect3DRMFrame\\:\.AddTransform"§

IDirect3DRMFrame::AddTransform

HRESULT AddTransform(D3DRMCOMBINETYPE rctCombine,

D3DRMMATRIX4D r.X, rvY, rvZ)

,,,•Š,fff f,f fff•œ,'%o,,

• œ,,,D3DRM_OKŽ",,,,ff,•,•',^—,,,,œDirect3D•Žff,•' ,Ž,,,,

rctCombine

,, ,•Š,œ,•Š,œ, ,•-,Ž,œD3DRMCOMBINETYP—<œ,fff

rvX, rvY, rvZ

x y z•œ,^'• , 'ç,,

Ž',,,•Š, , IDirect3DRMFrame の

3D

xe "IDirect3DRMFrame\\:\.AddVisual"§

IDirect3DRMFrame::AddVisual

HRESULT AddVisual(LPDIRECT3DRMVISUAL lpD3DRMVisual);

ff f,fffff ffffff,'%o,,

• œ,,,D3DRM_OK Ž",,,,ff ,•, •',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,

lpD3DRMVisual

Direct3DRMVisualfffff,Ž,•

fffff ffffff, ffff,fffff,Š, ff f,fffff ffffff,'%o
,,,, ff f,ff ,'

xe "IDirect3DRMFrame\\:\.DeleteChild"§

IDirect3DRMFrame::DeleteChild

HRESULT DeleteChild(LPDIRECT3DRMFRAME lpChild);

ff f,Š', , ff f,Ž ,,,,,, ,,,Žff f œœ ffff,,,"Š,,

• `CE,,,D3DRM_OK Ž",,,ff ,•, ,•',^—
 ,,,, , Dirt3D•Žf f,•' ,Ž ,,,, ,`

lpD3DRMVisual

`,Direct3DRMVisualfffff,Ž,•,ffff`

```
xe "IDirect3DRMFrame\:\:GetChildren"$
IDirect3DRMFrame::GetChildren
HRESULT GetChildren(LPDIRECT3DRMFRAMEARRAY* lpChildren);
```

Direct3DRMFrameArray の

• `CE,,,D3DRM_OK Ž",,,ff ,•, ,•',^—
 ,,,, , Direct3D•Žf f,•' ,Ž ,,,, ,`

lpChildren

`CE, , CE,,,, —CE,Direct3DRMFrameArrayffff,“,ffff,ffff
 Ž Direct3DRMFrameArray, Š‘`

```
xe "IDirect3DRMFrame\:\:GetColor"$
IDirect3DRMFrame::GetColor
```

`D3DCOLOR GetColor();`

"IDirect3DRMFrame\:\:GetLights"\$

```
IDirect3DRMFrame::GetLights
HRESULT GetLights(LPDIRECT3DRMLIGHTARRAY* lpLights);
```

`Direct3DRMLightArrayfffff,CEŽ,ff ,CEfff,Ž", ,`

• `D3DRM_OK
 Direct3D•Žf f,•' ,Ž ,,,, ,`

lpLights

`CE, , CE,,,, —CE,Direct3DRMLightArrayffff,“,ffff,ffff
 Ž IDirect3DRMLightArray`

```
xe "IDirect3DRMFrame\:\:GetMaterialMode"$
```

```
IDirect3DRMFrame::GetMaterialMode
```

`D3DRMMATERIALMODE GetMaterialMode();`

`f f,fffff f,Ž`

• の D3DRMMAentation"\$

IDirect3DRMFrame::GetOrientation

HRESULT GetOrientation(LPDIRECT3DRMFRAME lpRef, LPD3DVECTOR lprvDir, LPD3DVECTOR lprvUp);

ff□f,☒,,ž“,,,□,,•☒,□ž’,,,ž□ff□f,,`“,•,,,□

- □☒,, D3DRM_OK
- Direct3D の

lpRef

,Direct3DRMFrameffffff,ž,•

lprvDir, lprvUp

ff f,zž,yž,•☒,ž,ž,D3DVECTOR “

ž IDirect3DRMFrame::SetOrientation

xe "IDirect3DRMFrame\:\GetParent"\$

IDirect3DRMFrame::GetParent

HRESULT GetParent(LPDIRECT3DRMFRAME* lpParent);

☒ ,ff f, ff f,ž“,,,

- ☒,,,D3DRM_OK
- Direct3D•žf f,•’ ,ž ,,,, •’,^—,,,,,

lpP□’,,,□

xe "IDirect3DRMFrame\:\GetPosition"\$

IDirect3DRMFrame::GetPosition

HRESULT GetPosition(LPDIRECT3DRMFRAME lpRef, LPD3DVECTOR lprvPos);

ž’ ,,,,ž□ff□f,,,□ff□f,`^’,ž“,,,□,, のののの

lprvPos ,,,

- ☒,,,D3DRM_OK ž”,,,ff ,•, •’,^—,,,,,
- Direct3D•žf f,•’ ,ž ,,,,

lpRef

ž ,,—,,,,Direct3DRMFrameffffff,ž,•

lprvPos

ff f,^,ž,ž,D3DVECTOR “

ž□□IDirect3DRMFrame::SetPosition

xe "IDirect3DRMFrame\:\GetRotation"\$

IDirect3DFrame::GetRotation
 HRESULT GetRotation(LPDIRECT3DFRAME lpRef, LPD3DVECTOR
 lprvAxis,

LPD3DVALUE lprvTheta);

ff f, % "Ž", , % "Ž",

• D3DRM_OK
 Direct3D の

lpRef

Direct3DFrame

lprvAxis

D3DVECTOR

lprvTheta

の

IDirect3DFrame::SetRotation, Transformations

xe "IDirect3DFrame\:\GetScene"§

IDirect3DFrame::GetScene

HRESULT GetScene(LPDIRECT3DFRAME lpRoot);*

• D3DRM_OK
 ,,,, Direct3DŽ f f, ,Ž
 ,,,,

lpRoot

f f, ^ff f,Ž,Direct3DFramefffff,Ž,Ž,ffff,,ffff

*xe "IDirect3DFrame\:
 \:GetSceneBackground"§*

IDirect3DRMFrame::GetSceneBackground
D3DCOLOR
GetSceneBackground();

f f, "Œ ,Ž“,,
• ,•,

xe "IDirect3DRMFrame\:
\:GetSceneBackgroundDepth"§
IDirect3DRMFrame::GetSceneBackgroundDepth
HRESULT
GetSceneBackgroundDepth(

LPDIRECTDRAWSURFACE *
lpDDSurface);

f f,Œ , "Œ “ffff,Ž“,,
•

Œ,,,D3DRM_OK Ž”,,,ff ,
•, •’,^—
,,,, Direct3D•Žf f,•’ ,Ž
,,,

lpDDSurface
Œ , "Œ “ffff,Ž,DirectDraw
f fff,ffff,“ffff,ffff
Ž IDirect3DRMFrame::SetSceneBackgroundDepth
xe "IDirect3DRMFrame\:
\:GetSceneFogColor"§
IDirect3DRMFrame::GetSceneFogColor
D3DCOLOR
GetSceneFogColor();

f f,fff, ,Ž“,,
• fff, ,•,

xe "IDirect3DRMFrame\:
\:GetSceneFogEnable"§

```

IDirect3DRMFrame::GetSceneFogEnable
BOOL GetSceneFogEnable();

```

```

„f f, Ć fff,—Ć,,,,,,*,
• fff,—
Ć,,,,TRUE ,,,,,,FALSE,*,

```

```

xe "IDirect3DRMFrame\:
\:GetSceneFogMode"§
IDirect3DRMFrame::GetSceneFogMode
D3DRMFOGMODE
GetSceneFogMode();

```

```

„f f,Ć ,fff f f,*,
•
Ć ,fff f f,Ž,D3DRMFOG
MODE—(Ć,fff,,,,*,

```

```

xe "IDirect3DRMFrame\:
\:GetSceneFogParams"§
IDirect3DRMFrame::GetSceneFogParams
HRESULT
GetSceneFogParams(D3DVALUE * lprvStart, D3DVALUE *
lprvEnd,
D3DVALUE * lprvDensity);

```

```

„f f,Ć ,fff fff f,Ž“,
•
Ć,,,D3DRM_OK Ž”,,,ff ,*
, •,^—
,,,, Direct3D•Žf f,•’ ,Ž
,,,

```

```

lprvStart, lprvEnd, lprvDensity
fff,Ž“ “ —“,Ž,Ž,•

```

```

xe "IDirect3DRMFrame\:

```

```
\:GetSortMode"§
IDirect3DRMFrame::GetSortM
ode
D3DRMSORTMODE
GetSortMode();
```

```
Žff f, —,,,,Ž,,•—
f f,Ž“,,,
• •—
f f,Ž,D3DRMSORTMODE
—Œ,fff,•,
```

```
Ž IDirect3DRMFrame::SetS
ortMode
xe "IDirect3DRMFrame\
\:GetTexture"§
IDirect3DRMFrame::GetTextur
e
HRESULT
GetTexture(LPDIRECT3DRMT
EXTURE* lplpTexture);
```

```
Ž’,,,,ff f,ffffff,Ž“,,,
•
Œ,,,,D3DRM_OK Ž”,,,,ff ,
•, •’,^—
,,,, Direct3D•Žf f,•’ ,Ž
,,,,
```

```
lplpTexture
ff f,ffffff,Ž,Dire
```



```
IDirect3DRMFrame::SetTexture
xe "IDirect3DRMFrame\\.:GetTextureTopology"§
IDirect3DRMFrame::GetTextureTopology
HRESULT GetTextureTopology(BOOL * lpbWrap_u, BOOL * lpbWrap_v);
```

- **D3DRM_OK**
Direct3D の

```

v の TRUE
    Ž   IDirect3DRMFrame::SetTex
    extureTopology
xe "IDirect3DRMFrame\
\::GetTransform"§
IDirect3DRMFrame::GetTransform
m
HRESULT
GetTransform(D3DRMMATRIX4
D rmMatrix);

```

⑨ Direct3D の

D3DRMMATRIX4D ㊦㊦

3D

xe "IDirect3DRMFrameLPDIRECT3DRMFRAME lpRef, LPD3DVECTOR lprvVel,

BOOL fRotVel);

Ž',,,Ž□ff□f,`,,□ff□f,`'',Ž'',□

- □Œ,,D3DRM_OK□Ž'',,,ff□,•,□•',^—,,,,, Direct3D の

lpRef

,,—,,,Direct3DRMFrameffffff,Ž,•

lprvVel

ff f,“,Ž,Ž,D3DVECTOR “,ffff

fRotVel

~ŽŒ,“,Ž“,, fffffff,%o““, —
“,,,,,,Ž,fff ,fff f,TRUE,Ž',,,ffffff,%o““,ŒŽ,Š,,,
Ž IDirect3DRMFrame::SetVelocity

xe "IDirect3DRMFrame\:\:GetVisuals"§

IDirect3DRMFrame::GetVisuals

HRESULT GetVisuals(LPDIRECT3DRMVISUALARRAY* lpplVisuals);

のの

- D3DRM_OK
Direct3D の

lpplVisuals

Direct3DRMVisualArray

xe "IDirect3DRMFrame\:\:GetZbufferMode"§

IDirect3DRMFrame::GetZbufferMode

D3DRMZBUFFERMODE GetZbufferMode();

Z

Z

• D3DRMZBUFFT3DRMFrame\ \:InverseTransform"§

IDirect3DRMFrame::InverseTransform

HRESULT InverseTransform(D3DVECTOR *lprvDst, D3DVECTOR *lprvSrc);

lprvSrcfffff,Ž,ffff,fff
f,•,

- Ć,,D3DRM_OK Ž”,,,ff,•,•’,^—
,,,, Direct3D•Žf f,•’,Ž ,,,

lprvDst
•ŠĆ%,Ž,Ž,D3DVECTOR “

lprvSrc
•Š,f f,,D3DVECTOR “

Ž□□IDirect3DRMFrame::Transform, 3D
xe "IDirect3DRMFrame\::Load"§
IDirect3DRMFrame::Load

HRESULT Load(LPVOID lpvObjSource, LPVOID lpvObjID,

LPVOID lpArgLTP);

Direct3DRMFramefffff,f□f,,□

- □Ć,,D3DRM_OK□Ž”,,,ff□,•,□•’,^—,,,,□□Direct3D•Žf□f,•’□,Ž□,,,,□

lpvObjSource
fffff,f□f,,,f□f□d3drmLOFlagsfff□f,Ž,f□f□fff,,,,□fff□

lpvObjID

‘,,^’,fff f,d3drmLOFlagsfff f,Ž,Ž•Žfff,,,%Ž,, D3DRMLOA
D_BYPOSITIONfff,Ž’,,, ,fff f,ffff,’ffffff’,Ž,DWORD’,f
fff,, ,fff f,,NULL,Ž’,,,,,,

d3drmLOFlags

f f fffff,Ž,D3DRMLOADOPTIONSĆ,’

d3drmLoadTextureProc

D3DRMLOADTEXTURECALLBACKf ffffŠ ,,, “,ff fff,•—
,,ffffff,—,fffff,“, ,,,Ć, ,,, ,fff f,,NULL

lpArgLTP

D3DRMLOADTEXTURECALLBACKf□ffffŠ□,“,, ffff□fff

○ **lpvObjSource**

xe "IDirect3DRMFrame\:\:LPDIRECT3DRMFRAME lpRef,

D3DRMFRAMECONSTRAINT rfcConstraint) ;

Ž',,,,Ž□ff□f,"Š,,□ff□f,-•ff□f,⊕,□—,.,.,.,□⊕,%",fff,,□
 • □⊕,,,D3DRM_OK□Ž",,,ff□,•,□•,^—,.,.,.,□□Direct3D•Žf□f,•□,Ž□,.,.,□

lpTarget, lpRef

→,,,Ž□,.,.,.,Direct3DRMFramefffff,Ž,•□□

rfcConstraint

D3DRMFRAMECONSTRAINT

xe "IDirect3DRMFrame\:\:Move"§

IDirect3DRMFrame::Move

HRESULT Move(D3DVALUE delta);

—,.,.,Š',.,.,,ff f,%o“,“,“—,,
 • ⊕,,,D3DRM_OK Ž",,,ff,•, •,^—
 ,.,., Direct3D•Žf f,•',Ž,.,.,

delta

• ,,"%,%o“,—

xe "IDirect3DRMFrame\:\:SetColor"§

IDirect3DRMFrame::SetColor

HRESULT SetColor(D3DCOLOR rcColor);

ff f, ,',,, D3DRMMATERIALMODERIALMODE
 —⊕,D3DRMMATERIAL_FROMFRAME の の の
 ,Ž—,.,.,
 • □⊕,,,D3DRM_OK□Ž *Direct3D*
 の

rcColor

の

IDirect3DRMFrame::GetColor, IDirect3DRMFrame::SetMaterialMode

xe "IDirect3DRMFrame\:\:SetColorRGB"§

IDirect3DRMFrame::SetColorRGB

```
HRESULT SetColorRGB(D3DVALUE rvRed, D3DVALUE rvGreen,
    D3DVALUE rvBlue);
```

D3DRMMATERIALMODE
D3DRMMATERIAL_FROMFRAME

```
•  D3DRM_OK  Ž” ,ff ,•, •’,^—
,,,,  rect3D•Žf f,•’ ,Ž ,,,,  
```

rvRed, rvGreen, rvBlue

```
ff f,„Š—\,0,,1,“^,,,,,,,,  
```

```
Ž IIDirect3DRMFrame::SetMaterialMode
```

```
xe "IIDirect3DRMFrame\::SetMaterialMode"$
```

```
IIDirect3DRMFrame::SetMaterialMode
```

```
HRESULT SetMaterialMode(D3DRMMATERIALMODE rmmMode);
```

のの

```
•  D3DRM_OK  Ž” ,ff ,•, •’,^—
,,,,  Direct3D•Žf f,•’ ,Ž ,,,,  
```

rmmMode

D3DRMMATERIALMODE—(Efff ,,,,

Ž IIDirect3DRMFrame::GetMaterialMode

```
xe "IIDirect3DRMFrame\::SetOrientation"$
```

IIDirect3DRMFrame::SetOrientation

HRESULT SetOrientation(LPDIRECT3DRMFRAME lpRef, D3DVALUE
rvDx,

D3DVALUE rvDy, D3DVALUE rvDz, D3DVALUE rvUx, D3DVALUE
rvUy,

D3DVALUE rvUz);

```
ff f,• z [rvDx, rvDy, rvDz] y [rvUx,
rvUy, rvUz]
```

• **D3DRM_OK**

Dx, rvDy, rvDz

ff

rvUx, rvUy, rvUz

ff

ffff,ff,ffff[0, 0, 1],ffff[0, 1, 0],ž,,,
[rvUx, rvUy, rvUz],[rvDx, rvDy, rvDz]
,•,□,□,□D3DRMERR_BADVALUEff',•,,,□,,^š,□□,□",,, [rvUx,
rvUy, rvUz]ffff,[rvDx, rvDy,rvDz]

IDirect3DRMFrame::GetOrientation

xe "IDirect3DRMFrame\:\SetPosition"§

IDirect3DRMFrame::SetPosition

HRESULT SetPosition(LPDIRECT3DRMFRAME lpRef, D3DVALUE rvX,
D3DVALUE rvY,
D3DVALUE rvZ);

ž ff f,,,ff f,“^, ', ff f, ž ff f,,,←,[rvX, rvY, rvZ]
,',",",
,^',", ff f,[0, 0, 0]

• **D3DRM_OK** ž",,,ff ,•, •',^—, Direct3D
○•' ,ž ,,,

lpRef

ž ,,,—,,,Direct3DRMFrameffffff,ž,• ,ffff

rvX, rvY, rvZ

○

IDirect3DRMFrame::GetPosition

xe "IDirect3DRMFrame\:\SetRotation"§

IDirect3DRMFrame::SetRotation

HRESULT SetRotation(LPDIRECT3DRMFRAME lpRef, D3DVALUE rvX,
D3DVALUE rvY,

D3DVALUE rvZ, D3DVALUE rvTheta);

$$\text{IDirect}ff \sqsubseteq f, ' \langle, ,, \sqsubseteq$$

- ```

, , • □ , , , IDirect3DFrame::AddRotationffff, ^ , , □ , , ff

```

```
HRESULT SetSceneBackgroundImage(LPDIRECT3DRMTEXTURE lpTexture);
```

```
f f,"E f f f,Ž',,
• ,,D3DRM_OK Ž",,,ff ,*, •',^—
,,,,, Direct3D•Žf f,•' ,Ž ,,,,
```

```
lpTexture
,, Direct3DRMTexture
○○○○
```

```
xe "IDirect3DHRESULT
SetSceneBackgroundRGB(D3DVALUE rvRed,
D3DVALUE rvGreen,
```

```
D3DVALUE rvBlue);
```

```
f f f,"E , , , ,
• E,,,D3DRM_OK Ž", , , ff ,*, Direct3D の
```

```
rvRed, rvGreen, rvBlue
```

```
の
```

```
xe "IDirect3DRMFrame\::SetSceneFogColor"
IDirect3DRMFrame::SetSceneFogColor
HRESULT SetSceneFogColor(D3DCOLOR rcColor);
```

```
f f,fff, , ',,
• E,,,D3DRM_OK Ž",,,ff ,*, •',^—
,,,,, Direct3D•Žf f,•' ,Ž ,,,,
```

```
rcColor
の
```

```
xe "IDirect3DRMFrame\::SetSceneFogEnable"§
IDirect3DRMFrame::SetSceneFogEnable
```

```
HRESULT
SetSceneFo. E,,,D3DRM_OK Ž",,,ff ,*, ,•',^
—,,,,, Direct3D•Žf f,•' ,Ž ,,,,
```

```

bEnable
fff,□,□,□'□

xe "IDirect3DRMFrame\\:SetSceneFogMode"$
IDirect3DRMFrame::SetSceneFogMode
HRESULT SetSceneFogMode(D3DRMFOGMODE rfMode);

```

- D3DRM\_OK  
Direct3D の

rfMode

```

ff f f,Ž,D3DRMFOGMODE—(E,fff,,,
Ž IDirect3DRMFrame::SetSceneFogParams
xe "IDirect3DRMFrame\\:SetSceneFogParams"$
IDirect3DRMFrame::SetSceneFogParams
HRESULT SetSceneFogParams(D3DVALUE rvStart, D3DVALUE rvEnd,
D3DVALUE rvDensity);

```

```

,,f f,(E ,fff fff f, ',,
fff□f□f,,□fff,ŠŽ,□—^□,,,□',□□□,fffœ%o
,œ,,fff,,,←—,□fff,□‘—“,’,,,←—,œ',,□
rvDensity
Ž□fff□f□f,,fff,—“□,,',,,,,,,,,,"^,,,,,,,,□
Ž□□D3DRMFOGMODE, IDirect3DRMFrame::SetSceneFogMode
xe "IDirect3DRMFrame\\:SetSortMode"$
IDirect3DRMFrame::SetSortMode
HRESULT SetSortMode(D3DRMSORTMODE d3drmSM);

```

- f f, —,,,Ž,,•—f f,fff,, ,ffff, %o— <ffffff,• ,,,,—,,,,,,
- (E,,,D3DRM\_OK Ž”,,,ff ,•, •,^—  
,,,, Direct3D•Žf f,•' ,Ž ,,,,

d3drmSM

```

•—f f,Ž',,D3DRMSORTMODE—(E,∅
D3DRMSORT_FROMPARENT,,,
Ž IDirect3DRMFrame::GetSortMode
xe "IDirect3DRMFrame\\:SetTexture"$
IDirect3DRMFrame::SetTexture

```

```
HRESULT SetTexture(LPDIRECT3DRMTEXTURE lpD3DRMTexture);
```

- `D3DRM_OK` の値が返された場合は、`Direct3D` の `3DRMMATERIAL_FROMFRAME`、`ffffff,fff`、`ffff,,,Ž`、`ff,f,ffff,-`、`Ž`、`ffff, NULL, Ž`、`Ž`、`IFFrame::GetTexture`、`IDirect3DRMFrame::SetMaterialMode`、`xe "IDirect3DRMFrame\:\:SetTextureTopology"`§

```
IDirect3DRMFrame::SetTextureTopology
```

```
HRESULT SetTextureTopology(BOOL bWrap_u, BOOL bWrap_v);
```

```
ff f ,ffffff, fffff ,^“ ,’<,,
```

- `Ž`、`D3DRM_OK`、`Ž`、`ff`、`’`、`’`、`—`、`Direct3D`、`Ž`、`f`、`’`、`Ž`、`,,,`

```
bWrap_u, bWrap_v
```

```
□’,,Ž,^□□,•Ž,,,•Ž,ffffff,fff,,,□‘%o,,^ TRUE
IDirect3DRMFrame::GetTextureTopology
xe "IDirect3DRMFrame\:\:SetVelocity"§
```

**IDirect3DRMFrame D3DVALUE rvY,  
D3DVALUE rvZ, BOOL fRotVel);**

```
Ž,,,ff□f,□Ž□ff□f,,,`“□’,,□IDirect3DRM::Tick,,,IDirect3DRMFrame::Move
ffff,Ž,□,□Ž,,,□ff□f,ffff[rvX, rvY, rvZ],,,,^“,,,□
```

- `D3DRM_OK`  
Direct3D の

```
lpRef
```

```
Ž ,,—,,,Direct3DRMFrameffffff,Ž,•
```

```
rvX, rvY, ,,, rvZ
```

```
ff f, ,“
```

```
fRotVel
```

```
^ŽŽ,“ , ’,, fffffff,%o““ , —
“ ,,,,,Ž’ ,fff ,fff f,TRUE,,, fffffff,%o““ ,ŽŽ,Š,,,
```

```
Ž IDirect3DRMFrame::GetVelocity
```

```
xe "IDirect3DRMFrame\:\:SetZbufferMode"§
```

```
IDirect3DRMFrame::SetZbufferMode
```

```
HRESULT SetZbufferMode(D3DRMZBUFFERMODE d3drmZBM);
```

Zffff□f□f,□' Z

• □œ,,,D3DRM\_OK□Ž”,,,ff□,•,□•’,^—  
,,,,□□Direct3D•Žf□f,•’□,Ž□,,□

## d3drmZBM

Zffff□f□f,Ž„D3DRMZBUFFERMODE  
—(œ,fff,,,□ffff’,D3DRMZBUFFER\_FROMPARENT,,,□  
Ž□□IDirect3DRMFrame::GetZbufferMode

xe "IDirect3DRMFrame\:\.Transform"§  
IDirect3DRMFrame::Transform  
HRESULT Transform(D3DVECTOR \*lpd3dVDst, D3DVECTOR \*lpd3dVSrc);

**lpd3dVSrc** □□□□□□□□□□□□□□□□□□□□□□  
□□□□□□□□□□□□ **lpd3dVDst** □□□□□□□□  
□□•,□□Direct3D•Žf□f,•’□,Ž□,,,,□

## lpd3dVDst

•Š□—,œ%„Ž,Ž,D3DVECTOR□`□  
lpd3dVSrc  
•Š□—,,,D3DVECTOR□`□  
Ž□□IDirect3DRMFrame::InverseTransform, 3D•Š  
IDirect3DRMLight  
ffff□fff, □œ IDirect3DRMLight のの  
IDirect3DRMLight Œ Š”,,,,Š—  
, IDirect3DRMLight,IDirect3DRMLightArrayfff ffff ,Ž ,,,,  
IDirect3DRMLightfff ffff ffff, ^%„ff f,•—,,,,,,,  
œ  
GetConstantAttenuation

*GetLinearAttenuation*

*GetQuadraticAttenuation*

*SetConstantAttenuation*

*SetLinearAttenuation*



SetQuadraticAttenuation

GetColor

SetColor

SetColorRGB

—E”<sup>^</sup>  
GetEnableFrame

*SetEnableFrame*

*GetType*

# SetPenumbra

# SetUmbra

IDirect3DRMLightfff□ffff,□,,,COMfff□ffff,“—  
 □IUnknownfff□ffff□ffff,Œ□,,□IUnknownfff□ffff,Ž,,,,ffff,ff□f,,□  
 AddRef

QueryInterface

Release

IDirect3DRMLightfff ffff, IDirect3DRMObjectfff ffff,,Ž,ffff,Œ  
 ,,  
 AddDestroyCallback

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

**Direct3DRMLight** □□□□□□□□  
**IDirect3DRM::CreateLight** □□□  
**IDirect3DRMLight::GetColor**  
**D3DCOLOR GetColor();**

CE□,Direct3DRMLightfffff,□,Ž“, ,□  
 • □,•,□

Ž□□IDirect3DRMLight::SetColor  
 xe "IDirect3DRMLight\:\GetConstantAttenuation"\$  
 IDirect3DRMLight::GetConstantAttenuation  
 D3DVALUE GetConstantAttenuation();

Direct3DRMLight の

• CE ’ ,•,

CE ’ ,,’”——“,CE<“,‰o<,—,, ,,,, CE ’ ,2,,, CE,<“,”•,‰o,,

Ž IDirect3DRMLight::SetConstantAttenuation

xe "IDirect3DRMLight\:\GetEnableFrame"\$

IDirect3DRMLight::GetEnableFrame

HRESULT GetEnableFrame(LPDIRECT3DRMFRAME \* lpEnableFrame);

CECE, ’ ,,,,—CE,ff f,Ž“,,,

- `D3DRM_OK`, `Direct3D`, `Žf f,•' ,Ž ,,,,`, `•, •',^—`

**lpEnableFrame**

`CE□,Direct3DRMFrame`

□□

**IDirect3DRMLight::SetEnat::GetLinearAttenua  
tion**

**D3DVALUE GetLinearAttenuation();**

`CE, 1ŽCE□CE□, Ž", , □`

- `1ŽCE□CE□,•,□`

`Ž□□IDirect3DRMLight::SetLinearAttenuation`

`xe "IDirect3DRMLight\:\GetPenumbra"`

`IDirect3DRMLight::GetPenumbra`

`D3DVALUE GetPenumbra();`

`ffffff,^%o•,Š“,Ž“,,,`

- `^%o•,•,•,`

`Ž IDirect3DRMLight::SetPenumbra`

`xe "IDirect3DRMLight\:\GetQuadraticAttenuation"§`

`IDirect3DRMLight::GetQuadraticAttenuation`

`D3DVALUE GetQuadraticAttenuation();`

`CE,2ŽCE CE ,Ž“,,,`

- `2ŽCE CE ,•,`

`Ž IDirect3DRMLight::SetQuadraticAttenuation`

•  
.

**`CE□,Direct3DRMLightffffff,“^,Ž“,,,□`**

- `“^,Ž,‘,•,□`

`Ž□□IDirect3DRMLight::SetRange`

`xe "IDirect3DRMLight\:\GetType"§`

```
IDirect3DRMLight::GetType
D3DRMLIGHTTYPE GetType();
```

- D3DRMLIGHTTYPE

```
IDirect3DRMLight::SetType
xe "IDirect3DRMLight\\.\:GetUmbra"$
IDirect3DRMLight::GetUmbra
D3DVALUE GetUmbra();
```

Direct3DRenderLightfffff,-,Š“,Ž“,,  
• -,Š“,;

```

Ž IDirect3DLight::SetUmbra
xe "IDirect3DLight\::SetColor"§
IDirect3DLight::SetColor
HRESULT SetColor(D3DCOLOR rcColor);

```

•  $\check{Z} \in \mathcal{O}_{\text{D3DRM\_OK}}^{\text{ff}}$ ,  $\cdot$ ,  $\cdot^{\wedge}$ ,  $\text{Direct3D}$

```
rcColor
 CE, ..
 Ž□□IDirect3DLight::GetColor
 xe "IDirect3DLight\\:\\SetColorRGB"$
```

IDirect3DRMLight::SetCovBlue);

Ž',,,œœ,□,R, B, G,,,,',□',,□

- E,,,D3DRM OK□Ž",,,,ff□,•,□',^-,,,,,□□Direct3D•Žf□f,•'□,ž□,,,,□

```
rvRed, rvGreen, rvBlue
EE, ,
```

```
xe "IDirect3DRMLight\\:\\:SetConstantAttenuation"$
```

```
IDirect3DRMLight::SetConstantAttenuation
HRESULT SetConstantAttenuation(D3DVALUE rvAtt);
```

の

- D3DRM\_OK 値, ff, , , ^—  
Direct3D 値 f, , , ,

rvAtt

```
,,E E
E ' , , , ,E< ,%o< ,—, , , , E ' ,2,, E,< ,” ,’%o,,
Ž IDirect3DRMLight::GetConstantAttenuation
xe "IDirect3DRMLight\::SetEnableFrame"§
IDirect3DRMLight::SetEnableFrame
HRESULT SetEnableFrame(LPDIRECT3DRMFRAME lpEnableFrame);
```

- E, —E ff f, , ,  
D3DRM\_OK Ž , , ff , , , , ^— , , , , Direct3D

lpEnableFrame

ののの

## IDirect3DRMLight::GetELight::SetLinearAttenuation

```
HRESULT SetLinearAttenuation(D3DVALUE rvAtt);
```

E, 1ŽE E , , ,

- D3DRM\_OK Ž , , , ff , , , Direct3D の

rvAtt

```
IDirect3DRMLight::GetLinearAttenuation
xe "IDirect3DRMLight\::SetPenumbra"§
IDirect3DRMLight::SetPenumbra
HRESULT SetPenumbra(D3DVALUE rvAngle);
```

%o , ^%o•, Š , fff , ,

• D3DRM\_OK f ,•, •',^—  
 ,,,, Direct3D•Žf f,•' ,Ž ,,,,

rvAngle

□,,^%o•,Š“□,,Š“,^•,Š“,,‘,,,□“,,,,,,,,,□~%o•,Š“,—  
 •,Š“,t3DRMLight\:\:SetQuadraticAttenuation"§

IDirect3DRMLight::SetQuadraticAttenuation

HRESULT SetQuadraticAttenuation(D3DVALUE rvAtt);

CE, 2 ŽCE□CE□

• CE,,,D3DRM\_OK Ž”,,,ff ,•, •',^—  
 ,,,, Direct3D•Žf f,•' ,Ž ,,,,

rvAtt

□Light::SetRange

**HRESULT SetRange(D3DVALUE rvRange);**

CECE,“^,fff,,□CECE,□“^’,,,,fffff,,CE%o,—, ,□

• □CE,,,D3DRM\_OK□Ž”,,,ff□,•,□•',^—,,,,□□Direct3D•Žf□f,•',Ž□,,,□

rvRange

□,,“□fffff’,256,,,□

Ž□□IDirect3DRMLight::GetRange

xe "IDirect3DRMLight\:\:SetType"§

IDirect3DRMLight::SetType

HRESULT SetType(D3DRMLIGHTTYPE d3drmtType);

CECE,fff,• ,,

• CE,,,D3DRM\_OK Ž”,,,ff ,•, •',^—  
 ,,,, Direct3D•Žf f,•' ,Ž ,,,,

d3drmtType

D3DRMLIGHTTYPE 0 f f,,,,,Ž’,,

Ž IDirect3DRMLight::GetType

xe "IDirect3DRMLight\:\:SetUmbra"§

IDirect3DRMLight::SetUmbra

**HRESULT SetUmbra(D3DVALUE rvAngle);**

00

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

# GetEmissive

”ŽŽ□, <“

SetPower

SetSpecular

[illegible]



QueryInterface

Release

,,,IDirect3DRMMaterialfffIDirect3DRMObjtAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMMaterialffffff,IDirect3DRM::CreateMaterialfff,Œ,□,,,,Ž",,

xe "IDirect3DRMMaterial\\:GetEmissive"§  
IDirect3DRMMaterial::GetEmissive  
HRESULT GetEmissive(D3DVALUE \*lpr, D3DVALUE \*lpg, D3DVALUE \*lpb);

fffff.Ž“ , ‘Ž“,.,“ , ‘ , ,ffffff.”Ž,,Œ,“Ž,

• D3DRM\_OK Ž",,,ff,•,•',^—  
 ,,,, Direct3D Žf,f,•,Ž, ,,,

lpr, lpg, lpb

ffff, D3DRM,•,•,•,Ž, —, —,Ž,Ž,ffff

Ž IDirect3DRMMaterial::SetEmissive

xe "IDirect3DRMMaterial\:\:GetPower"\$

IDirect3DRMMaterial::GetPower

D3DVALUE GetPower();

Ž',,,fffff,,"ŽŽ< ",Ž",,□

• "ŽŽ< ",Ž',•,□

Ž IDirect3DRMMaterial::SetPower

xe "IDirect3DRMMaterial\:\:GetSpecular"\$

IDirect3DRMMaterial::GetSpecular

HRESULT GetSpecular(D3DVALUE \*lpr, D3DVALUE \*lpg, D3DVALUE  
 \*lpb);

fffff,"Žfffff, ,Ž“,,

• D3DRM\_OK Ž",,,ff Direct3D の

lpr, lpg, lpb

•,,"Žfffff, — , —,Ž,Ž,ffff

Ž IDirect3DRMMaterial::SetEmissive

HRESULT SetEmissive(D3DVALUE r, D3DVALUE g, D3DVALUE b);

fffff,•Ž□,□',,□

• D3DRM\_OK  
 Direct3D f,•',Ž ,,,

r, g, b

•Ž, , — —‘

Ž IDirect3DRMMaterial::GetEmissive

xe "IDirect3DRMMaterial\:\:SetPower"\$

IDirect3DRMMaterial::SetPower

HRESULT SetPower(D3DVALUE rvPower);

の

• 0E,,,D3DRM\_OK0Ž”,,,ff0,•,0’^—  
,,,,,00Direct•Žf0f,•’0,Ž0,,,0

## rvPower

0,,”ŽŽ00

Ž00IDirect3DRMMaterial::GetPower

xe "IDirect3DRMMaterial\\:SetSpecular"\$

IDirect3DRMMaterial::SetSpecular

HRESULT SetSpecular (D3DVALUE r, D3DVALUE g, D3DVALUE b);

0f, , ’,,

• 0E,,,D3DRM\_OK Ž”,,,ff ,•, •’^—  
,,,,, Direct3D•Žf f,•’ ,Ž ,,,,

r, g, b

”Žfffff, , — ,Š—‘

Ž IDirect3DRMMaterial::GetSpecular

IDirect3DRMMesh

ffff0fff, 0

IDirect3DRMMeshfff0ffff,fff,Ž—

.,0,,fffff,IDirect3DRMMeshfff0ffff0ffff,ffffff,.,0Š”,,,,,Š—

,00IDirect3DRMMesh,IDirect3DRMMeshBuilder

## GetGroupColoupCount

***fffff***

## GetGroupMaterial

SetGroupMaterial

*,,`*  
Scale

Translate

*ffffff•ž*  
GetGroupQuality

SetGroupQuality

*fffff□fffff*  
GetGroupMapping

SetGroupMapping

GetGroupTexture

SetGroupTexture

GetVertices

SetVertices

IDirect3DRMMeshffff ffff, ,,,COMfff ffff,“—  
IUnknownfff ffff ffff,Œ ,, IUnknownfff ffff,Ž,,,ffff,ff f,  
,  
AddRef

QueryInterface

Release

,,,□IDirect3DRMMeshfff□fff,IDirect3DRMObjectfff  
AddDestroyCallback

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMFrame

IDirect3DRM::CreateMesh

**xe "IDirect3DRMM unsigned fCount,**

unsigned vPerFace, unsigned \*fData, D3DRMGROUPINDEX \*returnId);

-, □□, ff□f%, □, , ff□f, ž•ž, ž“, , □

• □Œ,,D3DRM\_OK

Direct3D

vCount, fCount

ff f,š,,’“,–,

vPerFace

,,,–“, ,’“,ž, –“,,,’“, ff f’–,ž,’“, ,^,,,,, ,fff f,0,ž’,,,,,,,

fData

—

*f*□*f*,*ffff*□*vPerFaceffff*□*f*,',□',,,,,, ,*f f*,',  
*ff f*,'"—,,*ffffff*□*fff*,,,□*vPerFace*,0,,,  
□□□□,□,Ž,□□',—

□,,,,,,,□,,,,□*vPerFace*,0,□*ff*□*f*,□□,,—  
,ŽŠŒ,ŽŠŒ,,,□□□*f*□*f*,Ž,,,,,,□[ 3, *ffffff*,  
*ffffff*, *ffffff*, 4, *ffffff*, □□□□ , □□□□□  
□ Ž,•□,,*ffff*□

□,, '%o,,,*ff*□*f*,,□ ^%o,*ffff*,“□,—,,,□

- “
- *ffff*,,
- “Ž,,
- “<sup>1437</sup>—,Š<sup>14</sup>, ^'□—□□□,0

' ^', ^', □', , IDirect3DRMMesh::SetVertices



```

□□□□□□□□,□☒,•,,,□ff□f,“□,□,□,Ž,Ž,•□,ffff□,,,fff□f,,NULL,Ž',,,,,,□
vPerFace
ffff,□☒,•,,,□-“,,,‘“□,□,Ž,Ž,•□,ffff□,,,fff□f,,NULL,Ž',,,,,,□
fDataSize
fDataffff□f,Ž,ffff,‘,□•□,-‘,□,Ž,•☐☐
NULL

```

```
fData
```

```

☒,•,,, ff f,-
f f,Š”,,ffff,ffff,,f f,☒Ž, IDirect3DRMMesh::AddGroupffff,☒,
,,Ž’,,,,“,,, ,fff f,NULL,,, •—,ffff,fff,fDataSizefff f,•,,

```

```
xe "IDirect3DRMMesh\:\:GetGroupColor"§
```

```

IDirect3DRMMesh::GetGroupColor
D3DCOLOR GetGroupColor(D3DRMGROUPINDEX id);

```

```

ff f, ,Ž“,
• ☒,, ,Ž,D3DCOLOR’,•, ,,,, ,0,•,

```

```
id
```

## ffGroupColor, IDirect3DRMMesh::SetGroupColorRGB

```

xe "IDirect3DRMMesh\:\:GetGroupCount"§
IDirect3DRMMesh::GetGroupCount
unsigned GetGroupCount();

```

```
Direct3DRMMesh ☐☐
```

```
• ,, ,ff f, ,•, ,,,,0,•,
```

```
xe "IDirect3DRMMesh\:\:GetGroupMapping"§
```

```

IDirect3DRMMesh::GetGroupMapping
D3DRMMAPPING GetGroupMapping(D3DRMGROUPINDEX id);

```

□□□.ž”,,,0,•,□

Direct3DRMMesh 

- ## D3DRMRENDERQUALITY
- ### GetGroupQuality(D3DRMGROUPINDEX i....Group,Z,,□□,,,,,,,,,,,,,□

RGB Direct3DRMMesh  $\mathcal{O}\mathcal{O}$ 

- D3DRM\_OK Ǳ",,,ff ,• ,•,^—  
,,,, Direct3D•Ǳf f,• ,Ǳ ,,,

```

id
ff f,Ž•Ž ,Ž•Ž, IDirect3DRMMesh::AddGroup,Ž,, ,,,,,,,,,,
red, green, blue
ff f, , — ,Š—‘
Ž□□ IDirect3DRMMesh::GetGroupColor, IDirect3DRMMesh::SetGroupColor
xe "IDirect3DRMMesh\:\SetGroupMapping"§
IDirect3DRMMesh::SetGroupMapping
HRESULT SetGroupMapping(D3DRMGROUPINDEX id,
D3DRMMAPPING value);

```

**Direct3DRMMesh**fffff',ff□f,fffff,□',□fff  
 ff,□,,,,fffff,-,fff•”,,,ff□,•,□•',^—  
 ,,,,□□ **Direct3D•Žf□f,•'□,Ž□,,,□**

```

id
ff□f, Ž•Ž□,,Ž•Ž, IDirect3DRMMesh::AddGroup,Ž,,□□,,,,,,
,,,
value

```

#### D3DRMMAPPING

```

IDirect3DRMMesh::GetGroupMapping
xe "IDirect3DRMMesh\:\SetGroupMaterial"§
IDirect3DRMMesh::SetGroupMaterial
HRESULT SetGroupMaterial(D3DRMGROUPINDEX id,
LPDIRECT3DRMMATERIAL value);

```

```

Direct3DRMMeshfffff',ff□f,ff
• □œ,,,D3DRM_~OK□Ž”,,,ff□,•,□•',^—
,,,,□□ Direct3D•Žf□f,•'□,Ž□,,,□id
ff□f,Ž•Ž□,,Ž•Ž, IDirect3DRMMesh::AddGroup,Ž,,□□,,,,,,□
value
Direct3DRMMeshfffff,Š~•,, IDirect3DRMMaterialfffff,fffff□
IDirect3DRMMesh::GetGroupMaterial
xe "IDirect3DRMMesh\:\SetGroupQuality"§
IDirect3DRMMesh::SetGroupQuality
HRESULT SetGroupQuality(D3DRMGROUPINDEX id,
D3DRMRENDERQUALITY value);

Ž' ,,,ff f,ffffff•Ž, ',,

```

- $\mathbb{E},,,D3DRM\_OK \check{Z}'',,,ff \text{ ,}\bullet\text{ ,}\bullet'\text{ ,}\wedge\text{---}$   
 $,,,,, \quad Direct3D\bullet\check{Z}f \text{ f,}\bullet'\text{ ,}\check{Z} \text{ ,},,,$

id

$ff \text{ f,}\check{Z}\bullet\check{Z} \text{ ,},\check{Z}\bullet\check{Z},IDirect3DRMMesh::AddGroup,\check{Z},, \text{ ,},,,,\text{ ,},,,,\text{ ,},,,$

$D3DRMRENDERQUALITY\mathbb{E},\bullet,,\text{---}\langle\mathbb{E},'\text{ ,},,\text{'}\text{ ,}, \text{ ff}$

value

**IDirect3DRMMesh::GetGroupQuality**

xe "IDirect3DRMMesh\\:\:SetGroupTexture"§

**IDirect3DRMMRMTEXTURE value);**

**Direct3DRMMesh** $fffff\text{'},ff \square f,fffff,\check{S}\sim\bullet,,\square$

- $\square\mathbb{E},,,D3DRM\_OK\square\check{Z}'',,,ff\square,\bullet,\square\bullet'\text{ ,}\wedge\text{---},,,,\square\square Direct3D\bullet\check{Z}f\square f,\bullet'\square,\check{Z}\square,,, \square$

id

$ff\square f,\check{Z}\bullet\check{Z}\square,,\check{Z}\bullet\check{Z},IDirect3DRMMesh::AddGroup,\check{Z},,\square\square, \text{ ,},,,,\text{ ,},,,,\text{ ,},,,$   
 $,$

value

**Direct3DRMMesh**

**IDirect3DRMTexture の**

**IDirect3DRMMesh::GetGroupTexture**

xe "IDirect3DRMMesh\\:\:SetVertices"§

**IDirect3DRMMesh::SetVertices**

HRESULT SetVertices(D3DRMGROUPINDEX id, unsigned index,  
 unsigned count, D3DRMVERTEX \*values);

$\check{Z}',,,ff \text{ f,}\text{'}\wedge\text{ ,}\text{'}, \text{ ,},$

- $\square\mathbb{E},,,D3DRM\_OK\square\check{Z}'',,,ff\square,\bullet,\square\bullet'\text{ ,}\wedge\text{---},,,,\square\square Direct3D\bullet\check{Z} \text{ の}$

id

$\text{ の } \text{ の}$   
 $IDirect3DRMMesh::AddGroup,\check{Z},, \text{ ,},,,,\text{ ,},,,,\text{ ,},,,$

index

$\text{'}\wedge\text{ ,}\text{'}, \text{'},\check{Z}\text{'},\check{Z},\square valuesfff\square$

**count**

**index**

**values**

□□□□□□□□□□,,,,□□—•,ff□f,’“,ff□,,,,,,,,□

## Ž□ □ IDirect3DRMMesh::GetVertices

```
xe "IDirect3DRMMesh\\:\\:Translate"$
IDirect3DRMMesh::Translate
HRESULT Translate(D3DVALUE tx, D3DVALUE ty, D3DVALUE tz);
```

Direct3DRMMesh の実装

- D3DRM\_OK

 $tx, ty, tz$ 

Š'“^,x • y • z •,'%o,,fffff

IDirect3DRMMeshBuilder

ffff□fff,□fffffffffff,Œ,,,Ž,,,,IDirect3DRMMeshBuilderfff□ffff,fff  
f,Ž—,.,□,ffffff,□IDirect3DRMMeshBuilderfff□ffff□f Ɔ  
,,,Š—

```
, □ □ IDirect3DRMMesh, IDirect3DRMMeshBuilder f f f □ f f f f □ , Ž □ , ,
```

IDirect3DRMMeshBuilder ⑦

**GetColorSoace**

**AddFaces**

**CreateFace**

**GetFaceCount**

**GetFaces**

***f***  ***f***  
Load

*ffff*  
AddMesh

CreateMesh

*,,`*  
AddFrame

AddMeshBuilder

ReserveSpace

Save

Scale

SetMaterial

Translate

-□

AddNormal

GenerateNormals

SetNormal

◀◀

GetPerspective

SetPerspective

ffffff•Ž

GetQuality

SetQuality

fffff

GetTextureCoordinates

SetTexture

SetTextureCoordinates

*SetTextureTopology*

*AddVertex*

GetVertexColor

*GetVertexCount*

*GetVertices*

SetVertex

SetVertexColor

SetVertexColorRGB



IDirect3DRMMeshBuilderfff ffff, ,,,COMfff ffff,“—  
 IUnknownfff ffff ffff,Œ ,, IUnknownfff ffff,Ž,,,ffff,ff f,  
 ,  
 AddRef

*QueryInterface*

*Release*

,,, IDirect3DRMMeshBuilder  
 ,IDirect3DRMObjectfff ffff,,Ž,ffff,Œ ,,  
 AddDestroyCallback

Clone

*DeleteDestroyCallback*

*GetAppData*

GetClassName

GetName

*SetAppData*

*SetName*

*Direct3DRMMeshBuilder*fffff, □IDirect3DRM::CreateMeshBuilder

xe "IDirect3DRMMeshBuilder\:\AddFace"\$

*IDirect3DRMMeshBuilder::AddFace*

*HRESULT AddFace(LPDIRECT3DRMFACE lpD3DRMFace);*

*Direct3DRMMeshBuilder*

- D3DRM\_OK  
*Direct3D* の

lpD3DRMFace

の

のl の

xe "IDirect3DRMMeshBuilder\:\AddFaces"\$

**IDirect3DRMMeshBuilder::AddFaces**

**HRESULT AddFD3DVECTOR \*lpNormals,  
DWORD \*lpFaceData,**

LPDIRECT3DRMFACEARRAY\* lpD3DRMFaceArray);

- □, -, Direct3DRMMeshBuilderfffff, '%', □
- □œ,,D3DRM\_OK□Ž"„ff□,□•',^—,,,,□□Direct3D•Žf□f,•□,Ž□,,,,□

dwVertexCount

”“, □□

lpD3DVertices

D3DVECTOR

normalCount

lpNormals

$\mathcal{O}$  *D3DVECTOR*

## lpFaceData

のののnormalCount 0

,,," ,E,, , ,ffffff,ff,' ff, ,'"—,ffffff 2"—, —  
 ,,ffffff,Ž, fffffff,fff, E,,0, ',,,,,,,,,,

lplpD3DRMFaceArray

```

,, ,,,-,,ffff,Š",, IDirect3DRMFaceArrayfff ffff,,ffff,ffff

```

```
xe "IDirect3DRMMeshBuilder\:\:AddFrame"&
```

IDirect3DRMMeshBuilder::AddFrame

```
HRESULT AddFrame(LPDIRECT3D9FRAME lpD3D9Frame);
```

Direct3DRMMeshBuilderfffff,'%0,,

„—, , , œff f, • , , ž , , , , ,

```
xe "IDirect3DRMMeshBuilder\ :AddMesh"$
```

IDirect3DRMMeshBuilder::AddMesh

```
HRESULT AddMesh(LPDIRECT3DRMMESH lpD3DRMMesh);
```

```
ffff,Direct3DRMMeshBuilderffffff,'%,,□
```

- $\square \mathbb{E}_{,,,} \text{D3DRM\_OK} \square \check{Z}'' , , , f f \square , \square \bullet ' , \hat{^} _ , , , , \square \square \text{Direct3D} \bullet \check{Z} f \square f , \bullet '$

lpD3DRMMesh

```
xe "IDirect3DRMMeshBuilder\:\:AddMeshBuilder"§
```

## IDirect3DRMMeshBuilder::AddMeshBuilder

### **HRESULT** AddMeshBuilder(LPDIRECT3DRMMESHBUILDER lpD3DRMMeshBuild);

Direct3DRMMeshBuilder 〇〇 Direct3DRMMeshBuilderffffff, '%0,,

- `£,,D3DRM_OK Ž",,,ff ,. ,',^—`  
`,,,,, Direct3D•Žf f,',',Ž ,,,,`

lpD3DRMMeshBuild  
Direct3DRMMeshBuilderffffff,ffff ,ffffff,fffff,'%o,,  
,, —, ,,, のDirect3DRMMeshBuilderffffff,• ,,,Ž ,,,,,,,

xe "IDirect3DRMMeshBuilder\:\:AddNormal"§  
IDirect3DRMMeshBuilder::AddNormal  
int AddNormal(D3DVALUE x, D3DVALUE y, D3DVALUE z);

–□,Direct3DRMMeshBuilder

- ○

x, y, z  
○ x y z○

xe "IDirect3DRMMeshBuilder\:\:AddVertex"§  
IDirect3DRMMeshBuilder::AddVertex  
int AddVertex(D3DVALUE x, D3DVALUE y, D3DVALUE z);

Direct3DRMMeshBuilder  
•

x, y, zr::CreateFace  
HRESULT CreateFace(LPDIRECT3DRMFACE\*  
lpD3DRMFace);

"Ž,,,–□,□□,□Direct3DRMMeshBuilderffffff,'%s,,□  
• Œ,,,D3DRM\_OK Ž", の Direct3D  
Žf f,• ,Ž ,,,,

lpD3DRMFace  
,,,–,ffff,Š",,IDirect3DRMFacefff ffff,ffff,ffff

xe "IDirect3DRMMeshBuilder\:\:CreateMesh"§

```
IDirect3DRMMeshBuilder::CreateMesh
HRESULT CreateMesh(LPDIRECT3DRMMESH* lpD3DRMMesh);
```

**Direct3DRMMeshBuilder::CreateMesh**

## lpD3DRMMesh

**IDirect3DRMMesh**

```
xe "IDirect3DRMMeshBuilder\::GenerateNormals"$
IDirect3DRMMeshBuilder::GenerateNormals
HRESULT GenerateNormals();
```

```
Direct3DRMMeshBuilder
, — , * , , ,
• D3DRM_OK Ž” , , ff , * , * , ^ —
, , , , Direct3D Ž f , * , * , Ž , , ,
```

```
~ , , — , * , , , ff — , , ,
```

```
xe "IDirect3DRMMeshBuilder\::GetBox"$
IDirect3DRMMeshBuilder::GetBox
HRESULT GetBox(D3DRMBOX *lpD3DRMBox);
```

```
Direct3DRMMeshBuilder f f f f f , Š , < Ē ^
O
```

```
• D3DRM_OK
Direct3D O
```

```
lpD3DRMBox
```

```
D3DRMBOX
```

```
xe "IDirect3DRMMeshBuilder\::GetColorSource"$
IDirect3DRMMeshBuilder::GetColorSource
D3DRMCOLORSOURCE GetColorSource();
```

Direct3DRMMeshBuilder

- D3DRMCOLORSOURCE の

**IIDirect3DRMMeshBuilder**  
**int GetFaceCount();**

**Direct3DRMMeshBuilder** *fffff*, -, □, Ž“, ,, □

- -, □, •, □

```
xe "IDirect3DRMMeshBuilder\:\:GetFaces"$
IDirect3DRMMeshBuilder::GetFaces
HRESULT GetFaces(LPDIRECT3DRMFACEDATA*
lpD3DRMFaceData);
```

Direct3DRMMeshBuilder *fffff*, -, Ž“, ,,

- D3DRM\_OK Ž“, ,,ff ,•, •', ^—  
,,,, Direct3D•Žf f,•' ,Ž ,,,

lpD3DRMFaceData

-, ffff, Ž, Ž, IDirect3DRMFaceData fff□ffff, ,fff, f

```
xe "IDirect3DRMMeshBuilder\:\:GetPerspective"$
IDirect3DRMMeshBuilder::GetPerspective
BOOL GetPerspective();
```

Direct3DRMMeshBuilder

- TRUE FALSE

```
xe "IDirect3DRMMeshBuilder\:\:GetQuality"$
```

IDirect3DRMMeshBuilder::GetQuality

D3DRMRENDERQUALITY GetQuality();

Direct3DRMMeshBuilder の

## • **fect3DRMMeshBuilder\: \:GetTextureCoordinates"\$**

### **IDirect3DRMMeshBuilder**

```

::GetTextureCoordinates
HRESULT GetTextureCoordinates(DWORD index, D3DVALUE *lpU,
 D3DVALUE *lpV);

```

Direct3DRMMeshBuilderfffff',Ž',,,□",fffff□•,Ž",,,□

• D3DRM\_OK ",,ff ,•, •',^—  
 ,,,, Direct3D•Žf f,•' ,Ž ,,,,

index

"",fffff

lpU, lpV

ffff, ☉,•,, "“,fffff •,Ž,Ž,•

Ž IDirect3DRMMeshBuilder::SetTextureCoordinates

xe "IDirect3DRMMeshBuilder\:\:GetVertexColor"\$

IDirect3DRMMeshBuilder::GetVertexColor

## **D3DCOLOR GetVertexColor..ffff,"□-□□- f□f,Ž“,,□**

• ☉☉,,D3DRM\_OK□Ž",,ff□,•,□•',^-,,,,□□Direct3D•Žf□f,•'□,Ž□,,,,□

vcount

'",□,Ž,Ž,•□,ffff□

vertices

Direct3DRMMeshBuilder の Ž,D3DVECTOR “,”—

ncount

— , ,Ž,Ž,•

normals

Direct3DRMMeshBuilderfffff,- ,Ž,Ž,D3DVECTOR “,”—

face\_data\_size

face\_datafff f,Ž,ffff,fff,Ž',• ,,,fff,DWORD""^,—  
 ,,,, ,fff f,NULL,Ž',,,,,,,

face\_data

## **Direct3DRMMeshBuilderfffff,- f□f□,,ULL,,,□-ffff,•— ,,,fff,face\_data\_sizefff□f,•,,□**

xe "IDirect3DRMMeshBuilder\:\:Load"\$

```
IDirect3DRMMeshBuilder::Load
HRESULT Load(LPVOID lpvObjSource, LPVOID lpvObjID,

 D3DRMLOADOPTIONS d3drmLOFlags,
 D3DRMLOADTEXTURECALLBACK d3drmLoadTextureProc, LPVOID
 lpvArg);
```

Direct3DRMMeshBuilderfffff,f f,,

• ĄE,,D3DRM\_OKŹ”,,,ff□,•,□•’,^—,,,,,□□Direct3D•Žf□f,•

lpvObjSource

f□f,,fffff,f□f□d3drmLOFlagsfff□f,Ž,f□f□fff,,,□ffff□ff□f□fff  
□ffff

lpvObjID

□ □ d3drmLOFlags □□□□□□□□□□□□

fff□f,,NULL,Ž’,,,,,,□

d3drmLOFlags

f□f□fffff,Ž,D3DRMLOADOPTIONSĄE,’□

d3drmLoadTextureProc

D3DRMLOADTEXTURECALLBACKf□ffffŠ□,,,□“,ff□fff,•—,,,fffff,—  
,,,fffff“,□,,,ĄE,□,,,□,fff□f,,NULL,Ž’,,,,,,□

lpvArg

D3DRMLOADTEXTURECALLBACKf□ffffŠ□,“,,,,□f

Ň lpvObjSource ,,,f f,, ,ffff,f f,,

xe "IDirect3DRMMeshBuilder\\:\\:ReserveSpace"

IDirect3DRMMeshBuilder::ReserveSpace

HRESULT ReserveSpace(DWORD vertexCount, *DWORD normalCount*,  
*DWORD faceCount*);

’“ — —, ,Ž’ Direct3DRMMeshBuilder —

^Š•,, ,,,, ffff,,ĄE—“,fff,Ž,,,,,,

• ĄE,,D3DRM\_OK Ž”,,,ff ,•, •’,^—

,,,, Direct3D•Žf f,•’ ,Ž ,,,

vertexCount, *normalCount*, *faceCount*

’“ — —,,,Š,“,,—^,



```
xe "IDirect3DRMMeshBuilder\:\Save"$
IDirect3DRMMeshBuilder::Save
HRESULT Save(const char * lpFilename,
 D3DRMXOFFFORMAT d3drmXOFFFormat,
 D3DRMSAVEOPTIONS ...—
 ,,,,□□Direct3D•Žf□f,•'□,Ž□,,,□
```

## color

—,□□

```
xe "IDirect3DRMMeshBuilder\:\SetColorRGB"$
IDirect3DRMMeshBuilder::SetColorRGB
HRESULT SetColorRGB(D3DVALUE red, D3DVALUE green,
 D3DVALUE blue);
```

```
Direct3DRMMeshBuilderfffff,,,—, Ž',,, , ',,
• Ē,,,D3DRM_OK Ž",,,ff ,•, •',^—
 ,,,, Direct3D•Žf f,•' ,Ž ,,,,
```

red, green, blue  
,, , — ,Š—‘

```
xe "IDirect3DRMMeshBuilder\:\SetColorSource"$
IDirect3DRMMeshBuilder::SetColorSource
HRESULT SetColorSource(D3DRMCOLORSOURCE source);
```

## Direct3DRMMeshBuilder

- D3DRM\_OK  
Direct3D の

source

D3DRMCOLORSOURCE

IDirect3DRMMeshBuilder::GetColorSource

```
xe "IDirect3DRMMeshBuilder\:\SetMaterial"$
```

**IDirect3DRMMeshBuilder::SetMaterial**

**HRESULT**

**SetMaterial(LPDIRECT3DR...,D3DRM\_OK Ž",,,  
,ff□,•,□•',^—,,,,,□□Direct3D•Žf□f,•'□,Ž□,,,,,□**

perspective

ffff,ffff□ffff,‰<•□,“—,,□□,TRUE□,,,,,,FALSE,Ž',,,□

xe "IDirect3DRMMeshBuilder\:\:SetQuality"\$

IDirect3DRMMeshBuilder::SetQuality

HRESULT SetQuality(D3DRMRENDERQUALITY quality);

Direct3DRMMeshBuilderff の

- **D3DRM\_OK**のDirect3D の

**quality**

D3DRMRENDERQUALITY

IDirect3DRMMeshBuilder::GetQuality

xe "IDirect3DRMMeshBuilder\:\:SetTexture"\$

IDirect3DRMMeshBuilder::SetTexture

HRESULT SetTexture(LPDIRECT3DRMTEXTURE lpD3DRMTexture);

Direct3DRMMeshBuilderffffff,,,,-,fffff, ',,

- E,,,D3DRM\_OK Ž", の  
,,,, Direct3D•Žf f,•' ,Ž ,,,

lpD3DRMTexture

- ,Direct3DRMTextureffffff

xe "IDirect3DRMMeshBuilder\:\:SetTextureCoordinates"\$

IDirect3DRMMeshBuilder

::SetTextureCoordinates

HRESULT SetTextureCoordinates(DWORD index, D3DVALUE u,  
D3DVALUE v);

Direct3DRMMeshBuilderffffff,Ž',,,“,fffff •, ',,

- `D3DRM_OK` の場合、`Direct3D` のエラーコードは `D3DRM_OK` になります。

## `IDirect3DRMMeshBuilder::GetTextureCoordinates`

```
xe "IDirect3DRMMeshBuilder\::SetTextureTopology"$
IDirect3DRMMeshBuilder::SetTextureTopology
HRESULT SetTextureTopology(BOOL cylU, BOOL cylV);
```

`Direct3DRMMeshBuilder`

- `Direct3DRMMeshBuilder::SetTextureTopology` の戻り値は、`D3DRM_OK` の場合、`Direct3D` のエラーコードは `D3DRM_OK` になります。

`cylU, cylV`

`uŽŒ, vŽŒ, ^fff uŽŒ, vŽŒ, %'^, Ž,, , TRUE ,,,,,, FALSE, Ž',,`

```
xe "IDirect3DRMMeshBuilder\::SetVertex"$
IDirect3DRMMeshBuilder::SetVertex
HRESULT SetVertex(DWORD index, D3DVALUE x, D3DVALUE y,
D3DVALUE z);
```

`Direct3DRMMeshBuilder` のエラーコードは、`D3DRM_OK` の場合、`Direct3D` のエラーコードは `D3DRM_OK` になります。

- `D3DRM_OK` の場合、`Direct3D` のエラーコードは `D3DRM_OK` になります。

## `IDirect3DRMMeshBuilder::SetVertexColor`

```
IDirect3DRMMeshBuilder::SetVertexColor
HRESULT SetVertexColor(DWORD index, D3DCOLOR color);
```

`Direct3DRMMeshBuilder` のエラーコードは、`D3DRM_OK` の場合、`Direct3D` のエラーコードは `D3DRM_OK` になります。

- `D3DRM_OK` の場合、`Direct3D` のエラーコードは `D3DRM_OK` になります。

`index`

`color`

`color`

`Ž` `IDirect3DRMMeshBuilder::GetVertexColor`

```
xe "IDirect3DRMMeshBuilder\::SetVertexColorRGB"$
IDirect3DRMMeshBuilder::SetVertexColorRGB
```

`IDirect3DRMMeshBuilder::SetVertexColorRGB`

HRESULT SetVertexColorRGB(DWORD index, D3DVALUE red, D3DVALUE green, D3DVALUE blue);

Direct3DRMMeshBuilderffffff, Ž',,,“, , ',,  
• □Œ,,,D3DRM\_OK□Ž”,,,ff□,•,□•',^ Direct3D の

index

red, green, blue

□□□ □□□□□,Š—‘□

xe "IDirect3DRMMeshBuilder\\.\:Translate"§  
IDirect3DRMMeshBuilder::Translate  
HRESULT Translate(D3DVALUE tx, D3DVALUE ty, D3DVALUE tz);

Direct3DRMMeshBuilder  
• D3DRM\_OK  
Direct3D の

tx, ty, tz

∅x y z

IDirect3DRMObject

Direct3DRM  
IDirect3DRMObject の  
IDirect3DRMObject の

,,,Š—  
, IDirect3DRMMaterialfff  
ffff ,Ž ,,,  
IDirect3DRMObjectfff ffff  
ffff, ^%,ff f,•—,,,,,  
ffff fffŒ—,f f  
GetAppData

**SetAppData**

•

**Clone**

—‘

**GetClassName**

**GetName**

**SetName**

”

**AddDestroyCallback**

**DeleteDestroyCallback**

**IDirect3DRMObject***fff ffff,*  
*,,,COMfff ffff;“—*  
**IUnknown***fff ffff ffff,Œ*  
*„ IUnknownfff ffff,Ž,,,*  
*ffff,ff f,,*  
**AddRef**

**QueryInterface**

## Release

Direct3DRMObjectfffff, “  
,QueryInterfaceffff,Œ, ,,,,  
,,Direct3DRMfffff,,,Ž“,,  
,,,, ,,,Direct3DRMfffff, I  
Direct3DRMObjectfff ffff  
ffff,Œ ,,

xe "IDirect3DRMObject\  
\:AddDestroyCallback"§

IDirect3DRMObject::AddDestroyC  
allback

HRESULT  
AddDestroyCallback(D3DRMO  
BJECTCALLBACK lpCallback,  
LPVOID lpArg);

fffff,”Š,,,,,Œ, ,,Š ,“,,

- D3DRM\_OK

Direct3D の

*lpCallback*

**lpArg**

fffff,“,,,,,,

のの

xe "IDirect3DRMObject\  
\:Clone"§

IDirect3DRMObject::Clone

```
HRESULT Clone(LPUNKNOWN pUnkOuter, REFIID riid, LPVOID
*ppvObj);
```

- D3DRM\_OK

, •, ^—  
 ,,,, Direct3D•Žf f,• ,Ž ,,,

pUnkOuter  
 COM %o<”,‘%o,,,,,’ ,fff f  
 riid  
 • ,ffffff,Ž•Ž  
 ppvObj  
 ffff, Œ,•,,, fffffff,• ,Š”,fff  
 f

```
xe "IDirect3DRMObject\:

\DeleteDestroyCallback"§

IDirect3DRMObject::DeleteDestro

yCallback

HRESULT DeleteD
```

```
estroyCallback(D3DRMOBJECTCALLBACK d3drmObjProc,
 LPVOID lpArg);
```

**IDirect3DRMObject::AddDestroyCallback**

• **D3DRM\_OK**の  
,,',D3DRMOBJECTCALLBACKf ffffŠ  
**lpArg**

f ffffŠ,,',ffff ffff', f f, ffff

xe "IDirect3DRMObject\:\:GetAppData"§

IDirect3DRMObject::GetAppData

DWORD GetAppData();

ffff,, ffff fff—,32fff f f,Ž“, fffff',0,,  
• ffff

**IDirect3DRMObject::SetAppData**

**xe lpdwSize, LPSTR lpName);**

**fffff,f,f,f-,Ž“,,**

• —,D3DRM\_OKŽ“,,,,ff,•,•',^-,,,,Direct3D•Žff,•',Ž,,,□

lpdwSize

lpName

lpName

NULL, ,•Ž—  
,,ffff ,fff f, NULL, lpdwSizefff f,,•Ž—,•—,fff, ',,,

xe "IDirect3DRMObject\:\:GetName"§

IDirect3DRMObject::GetName

HRESULT GetName(LPDWORD lpdwSize, LPSTR lpName);



- `HRESULT D3DRM_OK` の `Direct3D` の `ff, - , Ž, NULL, , , Ž` —  
`„ffff, fff, NULL, , , lpdwSize fff, , Ž, —, fff, , , ,`  
`Ž IDirect3DRMObject::SetName`  
`xe "IDirect3DRMObject\::SetAppData"$`  
`IDirect3DRMObject::SetAppData`  
`HRESULT SetAppData(DWORD ulData);`

の 32 , ,

- `„, , D3DRM_OK Ž” , , ff , , , , ^` —  
`„ , , Direct3D Ž f , , , Ž , , ,`

`ulData`  
`ffffff, Š” , , f f , f f`  
`Ž IDirect3DRMObject::GetAppData`  
`xe "IDirect3DRMObject\::SetName"$`  
`IDirect3DRMObject::SetName`  
`HRESULT SetName(const char * lpName);`

`ffffff, - , , ,`

- `„, , D3DRM_OK` の  
`Direct3D` の

## lpName

`ff, , Š%o, , , IDirect3DRMShadow fff, ffff, ffff, Ž` —  
`„, ffff, ffff, IDirect3DRM::CreateShadow ffff, „, , , , , , Š% , —`  
`„ , , , Š% , — , , , IDirect3DRM::CreateObject ffff, „, ,`

の `IDirect3DRMShadow fff ffff ffff, ffff, , , Š” , , , Š` —  
`„ , IDirect3DRMShadow fff ffff, Ž , , ,`  
`IDirect3DRMShadow` **Init**

## Release

`„, IDirect3DRMShadow fff, ffff, IDirect3DRMObject fff, ffff, , Ž, ffff, „, ,`  
`AddDestroyCallback`

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMShadowfffff, IDirect3DRM::CreateShadow

xe "IDirect3DRMShadow\\:\\Init"§

**IMLight, D3DVALUE px, D3DVALUE py,**  
D3DVALUE pz, D3DVALUE nx, D3DVALUE ny, D3DVALUE nz) ;

Direct3DRMShadowfffff, §%, , □  
• Œ,, D3DRM\_OK  
Direct3D•Žf f,•’ ,Ž ,,,,

lpD3DRMVisual

%o%o•,,Direct3DRMVisualfffff

lpD3DRMLight

%o%o,Œ,,ŒŒ                   Direct3DRMLight

**px, py, pz**

のの

**nx, ny, nz**

の                   の ff□ffff,ffff,Ž—  
 „□fffff,Œ,fffff”—

„□„fffff,□IDirect3DRMTexturefff□ffff□ffff,ffffff,,„□Š”,,  
 ,,,Š-,□□IDirect3DRMTexture

IDirect3DRMTexture                   %o,ff f,•—,,,,,,,,,

GetColors

SetColors

ff f

GetDecalOrigin

GetDecalScale

GetDecalSize

**GetDecalTransparency**

**GetDecalTransparentColor** rency

**SetDecalTransparentColor**

*ff* ☐ *f*  
**GetImage**

☐ Š%  
InitFromFile

InitFromResource

InitFromSurface

Changed

GetShades

SetShades

IDirect3DRMTexturefff ffff, ,,,COMfff ffff,“—  
 IUnknownfff ffff ffff,Œ „ IUnknownfff ffff,Ž,,,ffff,ff f  
 ,,  
 AddRef

QueryInterface

Release

□□□□IDirect3DRMTextu

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName

SetAppData

SetName

Direct3DRMTexture IDirect3DRM::CreateTexture ∅ Ž“,,

xe "IDirect3DRMTexture\\:\:Changed"§  
IDirect3DRMTexture::Changed  
HRESULT Changed(BOOL bPixels, **BOOL bPalette**);

□ □  
□□□.□,•,□•',^—  
,,,,,□□**Direct3D•Žf□f,•'□,Ž□,,,□**

**bPixels**

ffff,•□,,,,,TRUE,□',,□  
bPalette  
ffff,•□,,,,,TRUE,□',,□

xe "IDirect3DRMTexture\\:\:GetColors"§  
IDirect3DRMTexture::GetColors  
DWORD GetColors();

ffffff,ffffff,Ž,,, ‘ ,Ž“,,  
• ,•,

,,ffff,□ffffff,□□,Ž,,,ff□f,□□,,,,□,,,,,fffff,□',,,, , ,•, ,,,, ' •,,, ,  
, IDirect3DRMTexture::SetColors  
,fffff,□,•□,,,Œ,□IDirect**3DRM::SetDefaultTextureColors**

## Direct3DTexture::GetDecalOrigin

```
HRESULT GetDecalOrigin(LONG * lpIX, LONG * lpIY);
```

 $f f \square f, \mathbb{E} \square, \mathbb{E}^{\prime \prime}, \check{Z}^{\prime \prime},, \square$ 

- `if (D3DRM_OK == hr) hr = Direct3D9::CreateDeviceEx(`

$|p|X, |p|Y$

 $ff$ 

IDirect3DTexture::SetDecalOrigin

```
xe "IDirect3DTexture\:\:GetDecalScale"&
```

IDirect3DTexture::GetDecalScale

```
DWORD GetDecalScale();
```

Ž',,,ff f,ff fff“ ,Ž“,,,

- $\mathcal{E}_{,,,ff} \quad ffff$  “ $\cdot, \check{Z}$ ” $_{,,, - , \cdot,}$

Ž□ *IDirect3DTexture::SetDecalScale*

```
xe "IDirect3DTexture\:\:GetDecaleight);
```

*ff* □ *f*, *fff*, *ž*“,, □

- `if (D3DRM_OK == hr) {`

lprvWidth, lprvHeight

ffff, ☐, •, •, •, •, ff☐f, •, ☐, •, ž, ž, •☐☐

### ŽŖI Direct3DTexture::SetDecalSize

```
xe "IDirect3DRMTexture\::GetDecalTransparency"§
```

IDirect3DTexture::GetDecalTransparency

```
BOOL GetDecalTransparency();
```

$$ff \quad f, \text{“} - \text{“} \quad , \check{Z} \text{“} , ,$$

- $ff \sqsubseteq f, “-” ,,,, \text{TRUE} \sqsubseteq ,,,, \text{FALSE}$

IDirect3DTexture::SetDecalTransparency

```
xe "IDirect3DTexture\::GetDecalTransparentColor"§
```

### IDirect3DTexture::GetDecalTransparentColor

**D3DCOLOR GetDecalTransparentColor();** $\mathcal{O}$

x"IDirect3DRMTexture\:\:GetImage"\$

IDirect3DRMTexture::GetImage

D3DRMIMAGE \* GetImage();

fffff,□□,—,,,ff□f,fffff,•,□  
• Ė□,fffff,□□,—,,,D3DRMIMAGE□``,,

xe "IDirect3DRMTexture\:\:GetShades"\$

IDirect3DRMTexture::GetShades

DWORD GetShades();

fffff,ffffff,, ,,,, ,Ž,,,%%%, ,Ž“,,  
• %%, ,•,

Ž IDirect3DRMTexture::SetShades

xe "IDirect3DRMTexture\:\:InitFromFile"\$

IDirect3DRMTexture::InitFromFile

HRESULT InitFromFile(const char \*filename);

Ž’,,, の

• □□□□□

D3DRM\_Offf,□IDirect3DRM::CreateObject,—

,,□□,,,,,,,,,□

Ž□□IDirect3DRMTexture::InitFromResource,

IDirect3DRMTexture::InitFromSurface

xe "IDirect3DRMTexture\:\:InitFromResource"\$

IDirect3DRMTexture::InitFromResource

HRESULT InitFromResource(HRSRC rs);

Ž’,,,ff f,,Direct3DRMTextureffffff, Š%,,

• Ė,,,D3DRM\_OK Ž”,,,ff ,•, •,^—  
,,,, Direct3D•Žf f,•’ ,Ž ,,,,

rs

IDirect3DRMTexture::InitFromFile,  
IDirect3DRMTexture::InitFromSurface

ff f,ffff,Ž’,,



## URFACE lpDDS);

```

Ž',,,DirectDrawf fff,f f,Ž—,,fffff,Š%o,,
• Œ,,,D3DRM_OK Ž",,,ff,•,•',^—,,,,,Direct3D•Žf

```

lpDDS

```

DirectDrawf fff
Ž IDirect3DRMTexture::InitFromFile,
IDirect3DRMTexture::InitFromResource
xe "IDirect3DRMTexture\\:SetColors"$
IDirect3DRMTexture::SetColors
HRESULT SetColors(DWORD ulColors);

```

```

fffff,f ,,,ffff,•—
,, fff ff fff,,,,
• Œ,,,D3DRM_OK Ž",,,ff ,•, •',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,

```

```

ulColors
fffff',8,,,
Ž IDirect3DRMTexture::GetColors
xe "IDirect3DRMTexture\\:SetDecalOrigin"$

```

## IDirect3DRMTexture::SetDecalScale

```

xe "IDirect3DRMTexture\\:SetDecalSize"$

```

```

IDirect3DRMTexture::SetDecalSize
HRESULT SetDecalSize(D3DVALUE rvWidth, D3DVALUE
rvHeight);

```

```

• Œ,,,D3DRM_OK Ž",,,ff ,•, •',^—
,,,, Direct3D•Žf f,•' ,Ž ,,,

```

```

rvWidth, rvHeight
ff f, ,•, ,fff •,Ž',, fffff,fff,[1,1],,,

```

Ž IDirect3DTexture::GetDecalSize  
xe "IDirect3DTexture\:\:SetDecalTransparency"\$  
IDirect3DTexture::SetDecalTransparency  
HRESULT SetDecalTransparency(BOOL bTransp);

□□□□の□□□□□,□,•',^—  
,,,,,□□Direct3D•Žf□f,•'□,Ž□,,,,□

## bTransp

ff□f,"-□,Ž,,,,,TRUE□•"-□,Ž,,,,,FALSE,Ž',,,□fffff',FALSE,,,□  
Ž□□IDirect3DTexture::GetDecalTransparency  
xe "IDirect3DTexture\:\:SetDecalTransparentColor"\$  
IDirect3DTexture::SetDecalTransparentColor  
HRESULT SetDecalTransparentColor(D3DCOLOR rcTransp);

ff f,"- , ' ,,  
• Ć,,,D3DRM\_OK Ž",,  
Direct3D•Žf f,•' ,Ž ,,,,

rcTransp  
,,,"- fffff,"- , ,,,  
Ž□□IDirect3DTexture::GetDecalTransparentColor

xe "IDirect3DTexture\  
\:SetShadesfffffŽ,fffff,Š□,Ž—,,,%%  
,□',□,□',,□

fffff,,,□fffff,%%%,□',□,,,,,□,-",fff,,□,,ffff,•-,,,,,□fff□ff□fff,,,,,□  
• D3DRM\_OK ,ff ,•, •',^—  
,,,,, Direct3D•Žf f,•' ,Ž ,,,,

ulShades  
%%%, , , ,',2,— ,,, fffff',16,,  
Ž IDirect3DTexture::GetShades  
IDirect3DRMUserVisual  
ffff□fff,□Direct3DRMUserVisualfffff,□Š%,,,,IDirect3DRMUserVisual  
ff,ffff,Ž—  
,,□ffff□fff,IDirect3DRM::CreateUserVisualfff,Ć,□,,,,□,,□Š%,•—  
,,,□□Š%,•—,,,□

IDirect3DRM::CreateObjectffff,Ⓔ,□,,ffff□ffffff,□□,,□□,,,,,□,,ffff,  
 □IDirect3DRMUserVisualfff□ffff□ffff,ffffff, の  
 IDirect3DRMUserVisual

**IDirect3DRMUserVisualfff□ffff,Initffff,ff**  
 □f,,□

IDirect3DRMUserVisual の COM IUnknown  
 IUnknown の

AddRef

QueryInterface

Release

IDirect3DRMUserVisual IDirect3DRMObject の

AddDestroyCallback

Clone

DeleteDestroyCallback

GetAppData

*GetClassName*

*GetName*

*SetAppData*

*SetName*

**Direct3DRMUserVisual** **IDirect3DRM::CreateUserVisual**

```
xe "IDirect3DRMUserVisu
 al\:\Init"$
 IDirect3DRMUserVisual::Init
 HRESULT
 Init(D3DRMUSERVISUALCAL
 LBACK d3drmUVProc, void *
 lpArg);

Direct3DRMUserVisualfffff, Š
%o,,
•
 Œ,,,D3DRM_OK Ž”,,ff ,
 •, •’,^
```

—, **Direct3D の**

**d3drmUVProc**

**D3DRMUSERVISUALCALLBACK**

**lpArg**

の

**IDirect3DRM::CreateUserVisual**

**IDir**

**ect3DRMUserVisual::Init,CE, ,**

•—

**,,,, IDirect3DRM::CreateObje**

**ctffff,,,f f fffff**

□f

**IDirect3DRMViewport**

*IDirect3DRMViewport*

の

**IDirect3DRMViewport**

**fff ffff,ffffff,,, Š”,,,,Š— GetBack**

**, IDirect3DRMViewport,IDirect**

**3DRMViewportArrayfff ffff ,**

**Ž ,,,,**

**IDirect3DRMViewportfff ffff GetFront**

**ffff, ^%,ff f,•—,,,,,,,**

**fff**

**GetCamera**

**GetPlane**

**SetCamera**

**SetBack**

**SetFront**

**ffffff ff f**

SetPlane

GetHeight

GetWidth

**GetField**

**SetField**

**Init**

の

**Clear**

Configure

ForceUpdate

GetDevice

**tion**

**SetProjection**

***ff* □ *fff***

**GetUniformScaling**

SetUniformScaling

・š

InverseTransform

Transform

IDirect3DRMViewportfff□ffff,□,,,,COMfff□ffff,“–  
IUnknownfff□ffff□ffff,⌘□,,□IUnknownfff□ffff のの

AddRef

QueryInterface

Release

IDirect3DRMViewport  
IDirect3DRMObjectfff ffff,,Ž,ffff,⌘ ,,  
AddDestroyCallback

Clone

DeleteDestroyCallback

GetAppData

GetClassName

GetName



SetAppData

SetName

*Direct3DRMViewport*fffff, □ IDirect3DRM::CreateViewport の

```
xe "IDirect3DRMViewport\:\:Clear"$
IDirect3DRMViewport::Clear
HRESULT Clear();
```

の

## • IDirect3DRMViewport\:\:Configure"\$

### IDirect3DRMViewport::Configure

```
HRESULT Configure(LONG lX, LONG lY, DWORD dwWidth,
DWORD dwHeight);
```

- D3DRM\_OK Ž”,,,ff ,\*, \*,^—  
,,,, Direct3D•Žf f,\*, Ž ,,,

lX, lY

ff□f□ の

**dwWidth, dwHeight**

の

**IX + dwWidth** □□□□□①□□□□□  
**D3DRMERR\_BADVALUE**,•,□

```
xe "IDirect3DRMViewport\:\:ForceUpdate"$
```

```
IDirect3DViewport::ForceUpdate
HRESULT ForceUpdate(DWORD dwX1, DWORD dwY1, DWORD dwX2,
 DWORD dwY2);
```

```

f f f f f , - ^ , < □ " , □ □ , , □ ž ' , , , - ^ , □ ž % , IDirect3DRenderDevice::Update
'''

```

- [illegible]

dwX1, dwY1

 $\hat{=}, \langle, \bullet$ 

dwX2, dwY2

$$-\hat{\phantom{a}}, \text{\textperthousand} \text{\textperthousand} \text{\textless}, \quad \bullet$$
$$ffff, \square \tilde{Z}', \dots, \langle \mathcal{E}, \dots, ' \dots, f \square ffff, \square \square, \dots, \square \square \square, \dots, \square fff$$

```
xe "IDirect3DRMViewport\:\:GetBack"§
```

IDirect3DRMViewport::GetBack

D3DVALUE GetBack();

$$ff \square f \square$$

- `□□□□□□□□ tCamera"§`

## IDirect3DRenderViewPort::GetCamera

```
HRESULT GetCamera(LPDIRECT3DRMFRAME *lpCamera);
```

$$ff \square f \square f, fff, \check{Z}^n, , \square$$

- $\square \mathbb{E}_{,,,} \text{D3DRM OK} \square \check{Z}''_{,,,} ff \square_{,,,} \bullet'_{, \wedge - ,,,,} \square \square \text{Direct3D の}$

## lpCamera

```
ff,•,Direct3DRenderFrameffffff,“,• ,,ffff
```

Ž IDirect3DRenderViewPort::SetCamera, Camera

```
xe "IDirect3D8RMViewport\:\:GetDevice">
```

IDirect3DRenderViewPort::GetDevice

*HRESULT GetDevice(LPDIRECT3DRMDEVICE \*lpD3DRMDevice);*

*„ffff,Ž“,*

## • □□□□□ D3DRM\_OKのDirect3D □□□□□ iewport\:\:GetDirect3DViewport"\$

*IDirect3DRMViewport::GetDirect3DViewport*

*HRESULT GetDirect3DViewport (LPDIRECT3DVIEWPORT \*  
lpD3DViewport);*

*Direct3DRMViewport Direct3Dff f f,Ž“,*

- *„D3DRM\_OK Ž“,„ff „, „,^—  
„„„ Direct3D•Žf f,•' ,Ž „„,*

*lpD3DViewport*

*Direct3DViewportffffff,ffff,“,ffff*

*xe "IDirect3DRMViewport\:\:GetField"\$*

*IDirect3DRMViewport::GetField*

**D3DVALUE GetField();**

## Ž□□ IDirect3DRMViewport::SetField, ViewinFrustum

## xe "IDirect3DRMViewport\:\:GetFront"\$

*IDirect3DRMViewport::GetFront*

*D3DVALUE GetFront();*

*ff□f□f,ffff□fffff□ff□f,^',Ž“,„□*

- *„Ž,•,□*

*Ž IDirect3DRMViewport::SetFront, Viewing Frustum*

*xe "IDirect3DRMViewport\:\:GetHeight"\$*

*IDirect3DRMViewport::GetHeight*

**DWORD GetHeight();**

*ff f f, „ffff'^,Ž“,*

- *„ffff'^,•,*

```
xe "IDirect3DRMViewport\:\:GetPlane"§
IDirect3DRMViewport::GetPlane
HRESULT GetPlane(D3DVALUE *lpd3dvLeft, D3DVALUE *lpd3dvRight,
```

**D3DVALUE \*lpd3dvBottom, D3DVALUE  
\*lpd3dvTop);**

ffff□ffffff□ff□

- D3DRM\_OKのDirect3D の

lpd3dvLeft, lpd3dvRight, lpd3dvBottom, lpd3dvTop

○○○

IDirect3DRMViewport::SetPlane

xe "IDirect3DRMViewport\:\:GetProjection"§

IDirect3DRMViewport::GetProjection

D3DRMPROJECTIONTYPE GetProjection();

f f,, • “%0,“Ž“%0,,,,,—  
,,,,,,

- D3DRMPROJECTIONTYPE  
—(E,fff,,,,,\*,

Ž IDirect3DRMViewport::Se  
tProjection

xe "IDirect3DRMViewport\  
\:GetUniformScaling"§

IDirect3DRMViewport::GetUni  
formScaling

BOOL GetUniformScaling();

fffff,fff,,’,•Žfff f,ff fff,,,  
,—,,, Ž“ ,Ž“,,

- ff f f,~,”—

```
,ff fff,,, ,TRUE ,,,,,,FA
LSE,•,
```

```
Ž IDirect3DRMViewport::Se
tUniformScaling
```

```
xe "IDirect3DRMViewport\
\GetWidth"§
```

```
IDirect3DRMViewport::GetWi
dth
```

```
DWORD GetWidth();
```

```
ff f f,•,ffff”Ž“,,
```

```
• •,ffff”^,•,
```

```
xe "IDirect3DRMViewport\
\GetX"§
```

```
IDirect3DRMViewport::GetX
```

```
LONG GetX();
```

```
ffff ,ff f fŽ“,x •ffff,Ž“
,,
```

```
• xfffff,•,
```

```
xe "IDirect3DRMViewport\
\GetY"§
```

```
IDirect3DRMViewport::GetY
```

```
LON
```

G GetY();

ののの y

- y

xe "IDirect3DRMViewport\:\:Init"§

IDirect3DRMViewport::Init

**HRESULT** Init(LPDIRECT3DRMDEVICE lpD3DRMDevice,

LPDIRECT3

**DRMFRAME**

**lpD3DRMFrameCamera,**

**DWORD xpos, DWORD ypos,**

**DWORD width, DWORD h**

eight);

**Direct3DRMViewport**

- *D3DRM\_OK*  
Direct3D の

,

**lpD3DRMDevice**

*„ff f f,Š~•,,,DirectD3DRM  
Devicefffff*

**lpD3DRMFrameCamera**

*„ff f f,Š~•,,,fff ff f*

xpos, ypos

*ff f f, <,x •,,,y •*

**width, height**

*ff f f,•, ,*

**xe "IDirect3DRMViewport\:  
InverseTransform"§**

IDirect3DRMViewport::Inverse  
Trans

form

**HRESULT InverseTransform(D3DVECTOR \* lprvDst, D3DRMVECTOR4D \*  
lprvSrc);**

**lprvSrc** □□□□□□□□□□□□□□□□□□□□□□□□  
□□□□□□□□□□ **lprvDst** □□□□□□□□  
,□,•,,,,,□□ **Direct3D•Žf**□**f,•'**□,Ž□,,,,,□

**lprvDst**

*fff,□Œ,•,,,□—Œ%,Ž,Ž,D3DVECTOR□`'□  
lprvSrc  
•Š,, ,D3DRMVECTOR4D□`'□*

**xe "IDirect3DRMViewport\:\:Pick"§**  
IDirect3DRMViewport::Pick  
HRESULT Pick(LONG lX, LONG lY,  
LPDIRECT3DRMPICKEDARRAY\* lplpVisuals);

□` , , , , • — , , , , f f f f f f □ Š ~ , , , , — , □ , f f f , Œ , , □ , □ Ž ` , , " — , , □ f

• Œ , , , D3DRM\_OK Ž " , , , f f , • , • ' , ^ —  
, , , , , Direct3D•Žf f , • ' , Ž , , , ,

lX, lY

*fffff,Ž,, , •*

lplpVisuals

CE, ,, CE,,, IDirect3DRMPickedArrayfff ffff,,—  
CE,ffff,“,ffff,,ffff

xe "IDirect3DRMViewport\:\:Render"§  
IDirect3DRMViewport::Render  
HRESULT Render(LPDIRECT3DRMFRAME lpD3DRMFrame);

ff fŠ‘, Ž‘,,,ff f f,ffffff,, Ž‘,,,ff f ,fffff, ,,‰  
,Š‘,ff f,,,ffffff,,,  
• CE,,,D3DRM\_OK Ž”,,,ff ,•, •‘,^—,,, Direct3D  
f f,•‘ ,Ž ,,,,

lpD3DRMFrame  
ffffff,,ff fŠ‘,Ž,Direct3DRMFrameffffff,Ž,• ,ffff

xe "IDirect3DRMViewport\:\:SetBack"§  
IDirect3DRMViewport::SetBack  
HRESULT SetBack(D3DVALUE rvBack);

ff f f,fff fffffff ff f,^‘, ’,,  
• CE,,,D3DRM\_OK Ž”,,,ff ,•, •‘,^—  
,,,,, Direct3D•Žf f,•‘ ,Ž ,,,,

rvBack  
fff fffffff ff f, ,^‘  
Ž IDirect3DRMViewport::GetBack, IDirect3DRMViewport::SetFront,  
Viewing Frustum

xe "IDirect3DRMViewport\:\:SetCamera"§  
IDirect3DRMViewport::SetCamera  
HRESULT SetCamera(LPDIRECT3DRMFRAME lpCamera);

’,,  
• □CE,,,D3DRM\_OK□Ž”,,,ff□,•,□•‘,^— Direct3D の

lpCamera  
Direct3DRMFrame fff,Ž,•  
,,ffff, ff f f,^‘ •CE CE,,Ž‘,,,fff ff f,“, , ’,, ff , fff ff f , ,z  
Ž•CE,‰,, ,yŽ, •CE,CE,,,  
Ž IDirect3DRMViewport::GetCamera, Camera  
xe "IDirect3DRMViewport\:\:SetField"§



```
IDirect3DRMViewport::SetField
HRESULT SetField(D3DVALUE rvField);
```

```
ff f f,ff ,ff ff, ',,
• D3DRM_OKŽ”,,,ff□,•,□,•,^—,,,,□□Direct3D•Žf□f,•,Ž
```

rvField

0.5 0 の

D3DRMERR\_BADVALUE

□□ IDirect3DRMViewport::SetFront

**IDirect3DRMViewport::SetFront**

```
HRESULT SetFront(D3DVALUE rvFront);
```

```
ff□f□f,ffff□fffff□ff□f,^',□',,,□
```

```
• D3DRM_OK
Direct3D の
```

rvFront

の

1.0 0

```
の',“,,, ,,ffff,D3DRMERR_BADVALUEff ,•,
Ž IDirect3DRMViewport::GetFront, Viewing Frustum
```

```
xe "IDirect3DRMViewport\::SetPlane"§
```

IDirect3DRMViewport::SetPlane

```
HRESULT SetPlane(D3DVALUE rvLeft, D3DVALUE rvRight, D3DVALUE
rvBottom,
D3DVALUE rvTop);
```

```
fff,zŽ,‘,,ff□f□f,fff,fff□f□Ž”,,,ff□,•,□,•,^—
,,,,□□Direct3D•Žf□f,•,Ž□,,,,□
```

rvLeft, rvRight, rvBottom, rvTop

```
ff□f□f,□□, x□• □□\, x□•□□□, y□•□□\, y□•□
```

```
, %
```

```
, '□,,, ff□f□f,Ž',, IDirect3DRMViewport::SetFieldffff,^,,,□,,,ffff,,,,□”^,
```

```
“—, ^', ff□f□f,Ž',,,,,,□,,,□•—,Ž□\,□',,□%-□-
```

```
,ffff□f□f,ffffffffff,,,,,□
```

Ž□□IDirect3DRMViewport::GetPlane, IDirect3DRMViewport::SetField

xe "IDirect3DRMViewport\:\.SetProjection"§

IDirect3DRMViewport::SetProjection

HRESULT SetProjection(D3DRMPROJECTIONTYPE rptType);

ff f f,Ž%fff, ',,

- Ą,,D3DRM\_OK Ž",,,ff ,\*, •',^—  
,,,, Direct3D•Žf f,•' ,Ž ,,,

rptType

D3DRMPROJECTIONTYPE—(Ą,fff,,,

Ž IDirect3DRMViewport::GetProjection

xe "IDirect3DRMViewport\:\.SetUniformScaling"§

IDirect3DRMViewport::SetUniformScaling

HRESULT SetUniformScaling(BOOL bScale);

•Žfff f,ffffff,fff,,',,ff fff,,,,,—,,, Ž“ , ',,

- Ą,,D3DRM\_OK ^—  
,,,, Direct3D•Žf f,•' ,Ž ,,,

bScale

,, Ž“ ,,,ff f,TRUE,Ž',,, ••Ą, '•Ą,“, ŽĄ ,Ž—

,, ,,,,,, •Žfff f,ffffff,^',,,,, ^, 0  
TRUE,,,

' ,,ffff, fffffff,ff f,,,,IDirect3DRMViewport::SetPlaneffff,,Ž—  
,,,

Ž IDirect3DRMViewport::GetUniformScaling

xe "IDirect3DRMViewport\:\.Transform"§

IDirect3DRMViewport::Transform

HRESULT Transform(D3DRMVECTOR4D \* lprvDst, D3DVECTOR \*  
lprvSrc);

lprvSrcfff f,Ž,ffff,f ff •,,fff f •,Š, Ą%,lprvDstfff f,•,

- Ą,,D3DRM\_OK Ž",,,ff ,\*, •',^—  
,,,, Direct3D•Žf f,•' ,Ž ,,,

lprvDst

•Š'□,□,,□^%0,•'Ž,□,,,,□Ą%0,ffff,Ž,“,,%0Ž□' ,,,□

μ §

IDirect3DRMWinDevice

```

ffff□fff,□ffffff□fff□ff,ffffff□fff□f,%",,,,,,IDirect3DRMW
inDevicefff□f の
IDirect3DRMWinDevicefff ffff ffff,ffffff,, Š",,,,Š—
, fffffŠ— ,Ž ,,,,
IDirect3DRMWinDevicefff ffff ffff, ^%o,ff f,•—,,,,,,,,
HandleActivate

```

HandlePaint

```

IDirect3DRMWinDevicefff ffff, ,,,COMfff ffff,“—
IUnknownfff ffff ffff,Œ ,, IUnknownfff ffff,Ž,,,,ffff,ff f
”
AddRef

```

QueryInterface

Release

```

Direct3DRMWinDevice □□□□□□□□
IID_IDirect3DRMWinDevice □□□□□
IDirect3DRM::QueryInterface □□□□□□□□□□
IDirect3DRM::CreateDevice,,Œ□,,,□

```

**xe "IDirect3DRMWinDevice\:  
 \:HandleActivate"\$**

```

IDirect3DRMWinDevice::HandleActivate
HRESULT HandleActivate(WORD wParam);

```

Windows の WM\_ACTIVATE

,,•,,,,,,,,

- D3DRM\_OK Ž”,,,ff ,•, •’,^—  
,,,,, Direct3D•Žf f,•’ ,Ž ,,,,

wParam

**fff□f□—**  
**fff□ff,DRMWinDevice::HandlePaint**  
**HRESULT HandlePaint(HDC hDC);**

Windows,WM\_PAINTfff□f,%,“,□hDCfff□f,□Windows,BeginPaintŠ□,“,,,,PAINTSTRUCT□`“,Ž“,,,,□,ffff,□fff□,□□,,,ff□f□f,Š`,—  
 ⑦

- D3DRM\_OK •, •’,^—  
,,,,, Direct3D•Žf f,•’ ,Ž ,,,,

hDC

ffff ffffff DC ,ffff

IDirect3DRMWrap

ffff fff, fff ffffff,,Ž,,,,,IDirect3DRMWrapfff ffff,ffff,Ž—  
 ,, ,fffff,IDirect3DRMWrapfff ffff ffff,fff Š”,,,,Š  
 —, IDirect3DRMWrapfff ffff ,Ž ,,,,  
 IDirect3DRMWrapfff ffff ffff, ^%o,ff f,•—,,,,,,  
 Š%o  
 Init

fff  
 Apply

ApplyRelative



— fffffff,ffff

Ž IDirect3D**RM::CreateWrap**

**xe "IDirect3D**RMWrap**\:\:ApplyRelative"**§

IDirect3D**RMWrap**::ApplyRelative

HRESULT *ApplyRelative*(LPDIRECT3D**RMFRAME** frame,  
LPDIRECT3D**RMOBJECT** mesh);

ffffff,“,fff, ’,, Š“ff f,f ff•Š, , Ž,fff,Ž ff f,◁f ff•Š,  
,,,

• □□□□□ **D3D**RM**\_OK**のDirect3D □□□□□

**apffffff**□

**Ž**□□ **IDirect3D**RM**::CreateWrap**

**xe "IDirect3D**RMWrap**\:\:Init"**§

IDirect3D**RMWrap**::Init

HRESULT Init(D3D**RMWRAP**TYPE d3drmw, LPDIRECT3D**RMFRAME** lpd3drmfRef,  
D3D**VALUE** ox, D3D**VALUE** oy, D3D**VALUE** oz,

D3D**VALUE** dx, D3D**VALUE** dy, D3D**VALUE** dz,

D3D**VALUE** ux, D3D**VALUE** uy, D3D**VALUE** uz,

D3D**VALUE** ou, D3D**VALUE** ov, **D3D**VALUE** su, D3D**VALUE** sv**);

Direct3D**RMWrap**ffffff, Š%,,,

• Œ,,,D3D**RM**\_OK Ž”,,,ff ,•, •’,^—  
,,,, Direct3D•Žf f,•’ ,Ž ,,,,

d3drmw

D3D**RMWRAP**TYPE のff,,,

lpd3drmfRef

„Direct3D**RMWrap**ffffff,Ž ff f,Ž,Direct3D**RMFrame**ffffff,ffff

ox, oy, oz

fff,Œ“

dx, dy, dz

fff,zŽ

ux, uy, uz

y

**ou, ov**

**su            sv**

```

□□ IDirect3DRMTOR min, max;
}D3DRMBOX;
typedef D3DRMBOX *LPD3DRMBOX;

```

```

IDirect3DRMMesh::GetBox, IDirect3DRMMeshBuilder::GetBoxffff,Ž",,<€"^^',<,
,□
min, max
<€"^^',<,,',□,,,,',D3DVECTOR□'',,,□
Ž□□D3DVECTOR, IDirect3DRMMesh::GetBox,
IDirect3DRMMeshBuilder::GetBox
xe "D3DRMIMAGE"$

```

D3DRMIMAGE

```

typedef struct _D3DRMIMAGE {
 int width, height;
 int aspectx, aspecty;
 int depth;
 int rgb;
 int bytes_per_line;
 void* buffer1;
 void* buffer2;
 unsigned long red_mask;
 unsigned long green_mask;
 unsigned long blue_mask;
 unsigned long alpha_mask;
 int palette_size;
 D3DRMPALETTEENTRY* palette;
}D3DRMIMAGE;
typedef D3DRMIMAGE, *LPD3DRMIMAGE;

```

**IDirect3DRM::CreateTexture** □□□□□□□□□□  
□□□□□□□□□□□□□□□□□□ **f**□**f**,•,□,□

**aspectx** ,, **aspecty**

”□•**Œffff,ffffff**”□

**depth**

**ffff**“,,,**fff**□□  
rgb  
,,**fff**,FALSE,□□□**ffff**,**ffff**,**ffffff**,Ž,□,,**fff**,TRUE,,,,□**ffff**,f□f%  
,,,RGB’,,,□  
bytes\_per\_line  
1**fff**“,,,**fff**□□,,’,4,“□,,,□  
buffer1  
**ffffff**,□,,,**fff**□□□,**ffff**□,**ffff**□  
buffer2  
2□**ffffff**,2”–,**ffff**ff□ NULL

red\_mask, green\_mask, blue\_mask, alpha\_mask

rgb,TRUE,,, ,,,**fff,ffff**, — •,**fff**,,, rgb,FALSE,,, **ffff**, —  
, — ,Š—‘**fff,fff**,,, ,,,‘,, SVGA**ffffff**,, Ž—,, —  
,“ ,64,,,,, ,,,**fff,0xfc,fff**,,,

palette\_size

**ffff**□**ff**  
palette  
rgb,FALSE, ,,,**fff,ffff ffff**,Ž,D3DRMPALETTEENTRY “,**fff**  
**f**,,,

Ž□□**IDirect3DRM::CreateTexture**, **IDirect3DRMTexture::GetImage**

xe "D3DRMLOADMEMORY"§

D3DRMLOADMEMORY

typedef struct \_D3DRMLOADMEMORY {

LPVOID lpMemory;

DWORD dSize;

} D3DRMLOADMEMORY, \*LPD3DRMLOADMEMORY;

**ffff fff,D3DRMLOAD\_FROMMEMORY**,Ž’,,IDirect3DRM::Load**fff**  
f ,,‘,Load**ffff** ,Œ, ,,,,f f,,,ff f,Ž,

lpMemory

dSize

f f,,,**fff ffff,fff,fff**^,Ž,

Ž Direct**3DRM::Load**, **IDirect3DRMAnimationSet::Load**,  
**IDirect3DRMFrame::Load**, **IDirect3DRMMeshBuilder::Load**,  
**D3DRMLOADOPTIONS**, **D3DRMLOADRESOURCE**



```
xe "D3DRMLOADRESOURCE"§
```

```
D3DRMLOADRESOURCE
```

```
typedef struct _D3DRMLOADRESOURCE {
D3DRMLOADRESOURCE,
*LPD3DRMLOADRESOURCE;
```

```
ffff□fff,D3DRMLOAD_
FROMRESOURCE,Ž',,IDirect3DRM::Loadffff□,,',Loadffff,,,,□,☒,□,,,,,f□f
,,,f□f,Ž,□
hModule
f□f,,,
の NULL
```

```
lpName
```

```
,-‘,Ž’,,
```

```
lpType
```

```
ff f,Ž•,,f f’◁,fff
```

```
lpName,lpTypeffff,□^f□f,0,□□□%o^f□f,□,,-‘,
□□Ž•Ž,,,Ž’,,,fff□f,fff,Ž,□,,,,,□,,,fff□f,NULL,
□,,•Ž—,,fff□ffff,,,□•Ž—
, “•Ž,ff f◁ # ,,,, Ž,,•Ž,ff f,-‘,fff,□□Ž•Ž,
Ž,10□□,•, ”#258”□ □
258,•,□-‘,’,,,□□Ž•Ž,Ž□,,,□ff□f,•—,,,fff,—
,☒,,,,,□□ffff,Ž““,,,□—,□,□
```

```
Ž□□IDirect3DRM::Load, IDirect3DRMAnimationSet::Load, IDirect3DRMFrame::Load,
IDirect3DRMMeshBuilder::Load, D3DRMLOADMEMORY, D3DRMLOADOPTIONS
xe "D3DRMPALETTEENTRY"§
D3DRMPALETTEENTRY
```

```
typedef struct _D3DRMPALETTEENTRY {
```

```
 unsigned char red;
```

```
 unsigned char green;
```

```
 unsigned char blue;
```

```
 unsigned char flags;
```

```
}D3DRMPALETTEENTRY;
```

```
typedef D3DRMPALETTEENTRY, *LPD3DRMPALETTEENTRY;
```

```

D3DRMIMAGE “,Ž—,,ff ffff,< ,, ,, “, D3DRMIMAGE の
rgbfff,FALSE,Ž’,,,,,,,,,—,,, TRUE,Ž’,,,,,,,, RGB’,—,,,
red, green, blue
ffff,’,& —,’,<,’ 0,,255,”^,’Ž’,,,,,,,,,
flags
,,,,,,,,Ž—,,,,,’<, D3DRMPALETTEFLAGS
のの

```

```

D3DRMIMAGE, D3DRMPALETTEFLAGS
xe "D3DRMPICKDESC"§
D3DRMPICKDESC
typedef struct _D3DRMPICKDESC}
D3DRMPICKDESC, *LPD3DRMPICKDESC;

```

```

IDirect3DRMPickedArray::GetPickffff,, , Ž“,,, □fff, ^’□-□ff□fŽ•Ž, Š“,, □
ulFaceIdx
Ž“,,, fffffff,-, fffffff□
lGroupIdx
Ž“,,, fffffff,ff□fŽ•Ž□
vPosition
Ž“,,, fffffff, ^’ , Ž, ’□,, ’, D3DVECTOR□`“,,, □
Ž□□D3DVECTOR, IDirect3DRMPickedArray::GetPick
xe "D3DRMQUATERNION"§

```

```

D3DRMQUATERNION
typedef struct _D3DRMQUATERNION {
 D3DVALUE s;
 D3DVECTOR v;
}D3DRMQUATERNION;
typedef D3DRMQUATERNION, *LPD3DRMQUATERNION;

```

```

IDirect3DRMAnimation::AddRotateKeyffff,—
,,,‰“,< ,, ,, Direct3D, ŠŠ ,,,,,Ž—,,,
Ž IDirect3DRMAnimation::AddRotateKey,
D3DRMQuaternionFromRotation, D3DRMQuaternionMultiply,
D3DRMQuaternionSlerp, D3DRMMatrixFromQuaternion
xe "D3DRMVECTOR4D"§
D3DRMVECTOR4D
typedef struct _D3DRMVECTOR4D {
 D3DVALUE x;
 D3DVALUE y;

```

```

 D3DVALUE z;

 D3DVALUE w;
}D3DRMVECTOR4D;

typedef D3DRMVECTOR4D, *LPD3DRMVECTOR4D;

```

の IDirect3DRMViewport, Ž —, , ,

x, y, z, w

D3DVALUE œ, ' • Š œ % œ, “, □

```

Ž □ □ IDirect3DRMViewport::Transform, IDirect3DRMViewport::InverseTransform
xe "D3DRMVERTEX"$
D3DRMVERTEX
typedef struct _D3DRMVERTEX{
 D3DVECTOR position;
 D3DVECTOR normal;

 D3DVALUE tu, tv;

 D3DCOLOR color;
} D3DRMVERTEX;

```

Direct3DRMMesh の

position

normal

の

tu, tv

”“, •, , , 'ffffff •

color

”“,

Ž IDirect3DRMMesh::GetVertices, IDirect3DRMMesh::SetVertices

—œ

```

xe "D3DRMCOLORSOURCE"$

```

D3DRMCOLORSOURCE

```

typedef enum _D3DRMCOLORSOURCE{

```

D3DRMCOLOR\_FROMFACE,

D3DRMCOLOR\_FROMVERTEX

```
} D3DRMCOLORSOURCE;
```

```
Direct3DRMMeshBuilder::SetColorSource(D3DRMCOLORSOURCE_
 D3DRMCOLOR_FROMFACE, IDirect3DRMMeshBuilder::GetColorSource(
 D3DRMCOLOR_FROMFACE, D3DRMCOLOR_FROMVERTEX
```

```
D3DRMCOLOR_FROMFACE
```

```
D3DRMCOLOR_FROMVERTEX
```

```
Direct3DRMMeshBuilder::GetColorSource
```

```
enum D3DRMCOMBINETYPE {
```

```
D3DRMCOMBINETYPE
```

```
typedef enum _D3DRMCOMBINETYPE {
 D3DRMCOMBINE_REPLACE,
 D3DRMCOMBINE_BEFORE,
 D3DRMCOMBINE_AFTER
} D3DRMCOMBINETYPE;
```

```
D3DRMCOMBINE_REPLACE
```

```
D3DRMCOMBINE_BEFORE
```

```
D3DRMCOMBINE_BEFORE
```

```
D3DRMCOMBINE_AFTER
```

```
D3DRMCOMBINE_AFTER
```

```
D3DRMCOMBINE_AFTER
```

```
IDirect3DRMFrame::AddRotation, IDirect3DRMFrame::AddScale,
IDirect3DRMFrame::AddTransform, IDirect3DRMFrame::AddTranslation
```

```
enum D3DRMFILLMODE {
```

```
D3DRMFILLMODE
```

```
typedef enum _D3DRMFILLMODE {
```

```
D3DRMFILL_POINTS = 0 * D3DRMLIGHT_MAX,
```

```
D3DRMFILL_WIREFRAME = 1 * D3DRMLIGHT_MAX,
```

```
D3DRMFILL_MASK = 7 * D3DRMLIGHT_MAX,
```

```
D3DRMFILL_MAX = 8 * D3DRMLIGHT_MAX
```

```

} D3DRMFILLMODE;

D3DRMRENDERQUALITY, ' < , Ž , , , - < , , , , □
D3DRMFILL_POINTS
" , , , " , , , , □ □ □ , " , , , , f □ f □
D3DRMFILL_WIREFRAME
fff □ f f □ f , " , □
D3DRMFILL_SOLID
- ` f f f f f f , " , □
D3DRMFILL_MASK
f f f , Ž - , , " , □
D3DRMFILL_MAX
" , , , , f □ f , □ ` ` □
Ž □ □ D3DRMLIGHTMODE, D3DRMSHADEMODE, D3DRMRENDERQUALITY
xe "D3DRMFOGMODE"$
D3DRMFOGMODE
typedef enum _D3DRMFOGMODE{
 D3DRMFOG_LINEAR,
 D3DRMFOG_EXPONENTIAL,
 D3DRMFOG_EXPONENTIALSQUARED
} D3DRMFOGMODE;

f f f , , , < - , □ , , , , □ f f f % , , , , , , ` , , , , , < , , , , Ž , □
D3DRMFOG_LINEAR
f f f %
§
f f , , , , , , ,
D3DRMFOG_EXPONENTIAL
f f f % , Ž Š “ , , , Ž , Ž , , ,
§

```

## D3DRMFOG\_EXPONENTIALSQUARED

```

の 2 の
μ §
, , , • Ž , , , , , , f f f ' , , , f f f f f , , , , ,
f f f , - “ Ž “ “ , IDirect3DRMFrame::SetSceneFogParams
e ∞
IDirect3DRMFrame::SetSceneFogMode,
IDirect3DRMFrame::SetSceneFogParams
xe "D3DRMFRAMECONSTRAINT"$
D3DRMFRAMECONSTRAINT
typedef enum _D3DRMFRAMECONSTRAINT {

```

## D3DR\_X

### } D3DRMFRAMECONSTRAINT;

```
Direct3DRMFramefffff,•Ž,,,,,<□",%"",Ž,<□,,□,-<Ⓔ, IDirect3DRMFrame::Loo
kAtfffff,,Ž-,,,□
D3DRMCONSTRAIN_Z
x,,,y%"",Ž-,,,□
D3DRMCONSTRAIN_Y
x,,,z%"",Ž-,,,□
D3DRMCONSTRAIN_X
y,,,z%"",Ž-,,,□
Ž□□IDirect3DRMFrame::LookAt
xe "D3DRMLIGHTMODE"§
D3DRMLIGHTMODE
typedef enum _D3DRMLIGHTMODE {
 D3DRMLIGHT_OFF = 0 * D3DRMSHADE_MAX,
 D3DRMLIGHT_ON = 1 * D3DRMSHADE_MAX,
 D3DRMLIGHT_MASK = 7 * D3DRMSHADE_MAX,
 D3DRMLIGHT_MAX = 8 * D3DRMSHADE_MAX
} D3DRMLIGHTMODE;

D3DRMRENDERQUALITYⒺ,'<Ž,,,—< ①
D3DRMLIGHT_OFF

 D3DRMLIGHT_ON

 ⒺⒺ,ff,,,
D3DRMLIGHT_MASK

 D3DRMLIGHT_MAX

 ⒺⒺf f, ‘
Ž□□D3DRMFILLMODE, D3DRMSHADEMODE,
D3DRMRENDERQUALITY

TYPE{

 D3DRMLIGHT_AMBIENT,

 D3DRMLIGHT_POINT,
 D3DRMLIGHT_SPOT,
 D3DRMLIGHT_DIRECTIONAL,
 D3DRMLIGHT_PARALLELPOINT
} D3DRMLIGHTTYPE;

IDirect3DRM::CreateLightfffff,Ⓔ,□,,□ⒺⒺfff,'<,,,□
D3DRMLIGHT_AMBIENT
```

```

fffffD3DRMLIGHT_POINT
ffffD3DRMLIGHT_SPOT
ffffffD3DRMLIGHT_DIRECTIONAL
ffffffD3DRMLIGHT_PARALLELPOINT
ffffD3DRMMATERIALMODE"$
D3DRMMATERIALMODE
typedef enum _D3DRMMATERIALMODE{
 D3DRMMATERIAL_FROMMESH,
 D3DRMMATERIAL_FROMPARENT,
 D3DRMMATERIAL_FROMFRAME
} D3DRMMATERIALMODE;
```

```

 IDirect3DRMFrame::GetMaterialMode
IDirect3DRMFrame::SetMaterialMode
D3DRMMATERIAL FROMMESH

```

```

“,,, ,,,fffff ’,,,
D3DRMMATERIAL_FROMPARENT
ffff□•,,,□ff□f,,£□,,,
D3DRMMATERIAL_FROMFRAME

```

```

0000000000000000 IDirect3DFrame::GetIMode
0000000000000000 xe "D3DRMPALETTEFLAGS"%

```

```
D3DRMPALETTEFLAGS
typedef enum _D3DRMPALETTEFLAGS {
 D3DRMPALETTE_FREE,
 D3DRMPALETTE_READONLY,
 D3DRMPALETTE_RESERVED
} D3DRMPALETTEFLAGS;
```

```
D3DRMPALETTEENTRY□`',□,ž-•-,',<,□
D3DRMPALETTE_FREE
ffff□,,,ffff,ž-,ž-,,,,□
D3DRMPALETTE_READONLY
&',,,,,,,,□ffff□,ž,,,,,,,,□
D3DRMPALETTE_RESERVED
ffff□,ž,,,,,,,,□
ž□□D3DRMPALETTEENTRY
xe "D3DRMPROJECTIONTYPE"S
```

## D3DRMPROJECTIONTYPE

```
typedef enum _D3DRMPROJECTIONTYPE{
 D3DRMPROJECT_PERSPECTIVE,
 D3DRMPROJECT_ORTHOGRAPHIC
} D3DRMPROJECTIONTYPE;
```

**Direct3DRMViewport** *Ž%0fff,‘<, , ,—<Œ*  
**IDirect3DRMViewport::GetProjection**, **IDirect3DRMViewport::SetProjection**

**D3DRMPROJECT\_PERSPECTIVE**

“Ž“%0

**D3DRMPROJECT\_ORTHOGRAPHIC**

## □□ **IDirect3DRMViewport::GetProjection,** **IDirect3DRMViewpRMRENDERQUALITY**

```
typedef enum _D3DRMSHADEMODE {
 D3DRMSHADE_FLAT = 0,
 D3DRMSHADE_GOURAUD = 1,
 D3DRMSHADE_PHONG = 2,
 D3DRMSHADE_MASK = 7,
 D3DRMSHADE_MAX = 8
} D3DRMSHADEMODE;

typedef enum _D3DRMLIGHTMODE {
 D3DRMLIGHT_OFF = 0 * D3DRMSHADE_MAX,
 D3DRMLIGHT_ON = 1 * D3DRMSHADE_MAX,

 D3DRMLIGHT_MASK = 7 * D3DRMSHADE_MAX,

 D3DRMLIGHT_MAX = 8 * D3DRMSHADE_MAX
} D3DRMLIGHTMODE;

typedef enum _D3DRMFILLMODE {
 D3DRMFILL_POINTS = 0 * D3DRMLIGHT_MAX,
 D3DRMFILL_WIREFRAME = 1 * D3DRMLIGHT_MAX,
 D3DRMFILL_SOLID = 2 * D3DRMLIGHT_MAX,
 D3DRMFILL_MASK = 7 * D3DRMLIGHT_MAX,
 D3DRMFILL_MAX = 8 * D3DRMLIGHT_MAX
} D3DRMFILLMODE;

typedef DWORD D3DRMRENDERQUALITY;

#define D3DRMRENDER_WIREFRAME
(D3DRMSHADE_FLAT+D3DRMLIGHT_OFF+D3DRMFILL_WIREFR
AME)
```



```
#define D3DRMRENDER_UNLITFLAT
(D3DRMSHADE_FLAT+D3DRMLIGHT_OFF+D3DRMFILL_SOLID)

#define D3DRMRENDER_FLAT
(D3DRMSHADE_FLAT+D3DRMLIGHT_ON+D3DRMFILL_SOLID)

#define D3DRMRENDER_GOURAUD
(D3DRMSHADE_GOURAUD+D3DRMLIGHT_ON+D3DRMFILLADE_PHONG+D3DRMLIGHT
_ON+D3DRMFILL_SOLID)

Direct3DRMMeshffffff,ff□ffff□f□f□□□f□□\,,,,,f□f,□□,,□
D3DRMSHADEMODE, D3DRMLIGHTMODE, D3DRMFILLMODE
ff□ffff□□□□\,,,,,f□f,Ž,□
D3DRMRENDER_WIREFRAME
•,,,•Ž,,□
D3DRMRENDER_UNLITFLAT
□□□□,ffff□f□f□f□f□
D3DRMRENDER_FLAT
ffff□f□f□f□f□
D3DRMRENDER_GOURAUD
f□f□□f□f□f□f□
D3DRMRENDER_PHONG
```

```

IDirect3DRMMesh::GetGroupQuality,
IDirect3DRMMesh::SetGroupQuality
xe "D3DRMSHADEMODE"§
D3DRMSHADEMODE
typedef enum _D3DRMSHADEMODE {
 D3DRMSHADE_FLAT = 0,
 D3DRMSHADE_GOURAUD = 1,
 D3DRMSHADE_PHONG = 2,
 D3DRMSHADE_MASK = 7,
 D3DRMSHADE_MAX = 8
} D3DRMSHADEMODE;

```

# D3DRMRENDERQUAL, D3DRMRENDERQUALITY

## xe "D3DRMSORTMODE"\$

```
D3DRMSORTMODE
typedef enum _D3DRMSORTMODE {
 D3DRMSORT_FROMPARENT,
 D3DRMSORT_NONE,
 D3DRMSORT_FRONTTOBACK,
 D3DRMSORT_BACKTOFRONT
} D3DRMSORTMODE;
```

f□f□,,,,,žff□f,•–,,,,,ž,□

D3DRMSORT\_FROMPARENT

žff□f,□ff□f,f□f□□,

D3DRMSORT\_NONE

D3DRMSORT\_FRONTTOBACK

D3DRMSORT\_BACKTOFRONT

**IDirect3DRMFrame::GetSortMode,**

**IDirect3DRMFrame::SetSortMode**

**xe "D3DRMTEXTUREQUALITY"§**

**D3DRMTEXTUREQUALITY**

typedef enum **\_D3DRMTEXTUREQUALITY**{

**D3DRMTEXTURE\_NEAREST,**

**D3DRMTEXTURE\_LINEAR,**

**D3DRMTEXTURE\_MIPNEAREST,**

**D3DRMTEXTURE\_MIPLINEAR,**

**D3DRMTEXTURE\_LINEARMIPNEAREST,**

**D3DRMTEXTURE\_LINEARMIPLINEAR**

**} D3DRMTEXTUREQUALITY;**

**IDirect3DRMDevice::SetTextureQuality □**

**IDirect3DRMDevice::GRE\_NEAREST**

**fffff□,□,<,fffff,“,,□**

D3DRMTEXTURE\_LINEAR

□,<,4,,ffff,□Œ,•Š,,□

D3DRMTEXTURE\_MIPNEAREST

D3DRMTEXTURE\_NEAREST,ž,,,,□fffff,`,,,“□,ffffff,ž–,,□

D3DRMTEXTURE\_MIPLINEAR

D3DRMTEXTURE\_ LINEAR,ž,,,,□fffff,`,,,“□,ffffff,ž–,,□

D3DRMTEXTURE\_LINEARMIPNEAREST

D3DRMTEXTURE\_ MIPNEAREST,ž,,,,□□,<,,,,,ffffffš,•Š,,□

D3DRMTEXTURE\_LINEARMIPLINEAR

D3DRMTEXTURE\_ MIPLINEAR,ž,,,,□□,<,,,,,ffffffš,•Š,,□

**xe "D3DRMUSERVISUALREASON"§**

**D3DRMUSERVISUALREASON**

```

typedef enum _D3DRMUSERVISUALREASON {
 D3DRMUSERVISUAL_CANSEE,
 D3DRMUSERVISUAL_RENDER
} D3DRMUSERVISUALREASON;

ffff,D3DRMUSERVISUALCALLBACKffffš,Œ,□,,—,'<,,□
D3DRMUSERVISUAL_CANSEE
f□f□ffffffffffff,f□f□f□,%ž□',,,,□f□ffffš□,TRUE,•,,,,,,,,,□
D3DRMUSERVISUAL_RENDER
f□ffffš□,f□f□fffff□fffff,fffff,,,,,,,,,□
ž□□D3DRMUSERVISUALCALLBACK
xe "D3DRMWRAPTYPE"$
D3DRMWRAPTYPE
typedef enum _D3DRMWRAPTYPE{
 D3DRMWRAP_FLAT,
 D3DRMWRAP_CYLINDER,
 D3DRMWRAP_SPHERE,
 D3DRMWRAP_CHROME
} D3DRMWRAPTYPE;

IDirect3DRM::CreateWrapffff,,,□□,,,Direct3DRMWrapfffff,fff,'<,,□,,I
Direct3DRMWrap::Initffff,Œ,□,,',□Direct3DRMWrapfffff,□š%,,,,,,—<Œ,ž
—,,,,,,,,□
D3DRMWRAP_FLAT
•-fff□
D3DRMWRAP_CYLINDER
%'fff□
D3DRMWRAP_SPHERE
<'fff□
D3DRMWRAP_CHROME
fff,□fffff,fffff□,"ž,,,,,fffff□•,š,",,,□
ž□□IDirect3DRM::CreateWrap, IDirect3DRMWrap::Init,
IDirect3DRMWrapfff□ffff
xe "D3DRMXOFFORMAT"$
D3DRMXOFFORMAT
typedef enum _D3DRMXOFFORMAT{
 D3DRMXOF_BINARY,
 D3DRMXOF_COMPRESSED,
 D3DRMXOF_TEXT
} D3DRMXOFFORMAT;

IDirect3DRMMeshBuilder::Saveffff,ž—,,,ffff□fff,'<,,□
D3DRMXOF_BINARY
ffff,ffff□ff□fff,,,□,,,fffff□',,,□
D3DRMXOF_COMPRESSED
Œ□,ff□f,,,,,□
D3DRMXOF_TEXT
ffff,ffff□ff□fff,,,□
D3DRMXOF_BINARY,D3DRMXOF_TEXT□',"ž,ž—,,,,,,,,,□
ž□□IDirect3DRMMeshBuilder::Save
xe "D3DRMZBUFFERMODE"$
D3DRMZBUFFERMODE
typedef enum _D3DRMZBUFFERMODE {
 D3DRMZBUFFER_FROMPARENT,

```

```

 D3DRMZBUFFER_ENABLE,
 D3DRMZBUFFER_DISABLE
 } D3DRMZBUFFERMODE;

Zffff,-E,,,,Ž,□
D3DRMZBUFFER_FROMPARENT
ff□f,□ff□f,,Zffff,□',E□,,□,,ffff□',,,□
D3DRMZBUFFER_ENABLE
Zffff,-E,,,□
D3DRMZBUFFER_DISABLE
Zffff,-E,,,□
Ž□□IDirect3DRMFrame::GetZbufferMode, IDirect3DRMFrame::SetZbufferMode
,,',E

```

```

xe "D3DRMANIMATIONOPTIONS"$
D3DRMANIMATIONOPTIONS
typedef DWORD D3DRMANIMATIONOPTIONS;
#define D3DRMANIMATION_CLOSED 0x02L
#define D3DRMANIMATION_LINEARPOSITION 0x04L
#define D3DRMANIMATION_OPEN 0x01L
#define D3DRMANIMATION_POSITION 0x00000020L
#define D3DRMANIMATION_SCALEANDROTATION 0x00000010L

#define D3DRMANIMATION_SPLINEPOSITION 0x08L

```

**IDirect3DRMAnimation::GetOptions**

**IDirect3DRMAnimation::SetOptions**

**D3DRMANIMATION\_CLOSED**

```

fff□fff,E,•,□□,,□□“,’,,,ŠŽ“,-,,f□f,,□•,,のののの
,,E,•,f□,——,,□f□f,,f ののの

```

**D3DRMANIMATION\_LINEARPOSITION**

```

fff fff,^', E, ',,,

```

**D3DRMANIMATION\_OPEN**

**1** 'Ž,,

**D3DRMANIMATION\_POSITION**

のの

,,,

**D3DRMANIMATION\_SCALEANDROTATION**

のの

',,,,•Š —, ,,,,

**D3DRMANIMATION\_SPLINEPOSITION**

```

fff fff,^',fffff, ',,,

```

xe "D3DRMCOLOMODEL"\$





**D3DRMCOLOMODEL**

**typedef D3DCOLOMODEL D3DRMCOLOMODEL;**

## D3DCOLORMODEL

□ □ D3DCOLORATIONS

```
typedef DWORD D3DRMLOADOPTIONS;
#define D3DRMLOAD_FROMFILE 0x00L
#define D3DRMLOAD_FROMRESOURCE 0x01L
#define D3DRMLOAD_FROMMEMORY 0x02L
#define D3DRMLOAD_FROMSTREAM 0x03L
#define D3DRMLOAD_BYNAME 0x10L
#define D3DRMLOAD_BYPOSITION 0x20L
#define D3DRMLOAD_BYGUID 0x30L
#define D3DRMLOAD_FIRST 0x40L
#define D3DRMLOAD_INSTANCEBYREFERENCE 0x100L
#define D3DRMLOAD_INSTANCEBYCOPYING 0x200L
```

**IDirect3DRM::Load**   
**IDirect3DRMAnimationSet::Load**   
**IDirect3DRMFrame::Load**   
**IDirect3DRMMeshBuilder::Load** 

 $f \square f \square fff$ 

## D3DRMLOAD\_FROMFILE

```

ffff,,f□f,,□,,,fffff□',,,,□
D3DRMLOAD_FROMRESOURCE
ff□f,,f□f,,□,,,fff,ž',,,□□□Loadffff,Ⓔ,□,,lpvObjSourceffff□f,□D3DRMLOADRE
SOURCE□`\",ž,,,,,,,,,□
D3DRMLOAD_FROMMEMORY
ffff,,f□f,,□,,,fff,ž',,,□□□Loadffff,Ⓔ,□,, lpvObjSource
fff□f,□D3DRMLOADMEMORY□`\",ž,,,,,,,,,□
D3DRMLOAD_FROMSTREAM
fff□f,,f□f,,□
ž•žfff
D3DRMLOAD_BYNAME

```

### D3DRMLoad BYPOSITION

$$\begin{array}{c} \mathbf{0} \mathcal{O} \\ \mathcal{O}_n - \\ , f f f f f f, f \quad f, , \quad f f f f \quad f f f, f f f f f f, ' , f f f f f f, \check{S}, , , , , , , \quad ' , f f f f f f, \check{S}, , , , , , , \end{array}$$

## D3DRMLOAD\_BYGUID

GUID ,Ž—,,ffffff,f f,,

## D3DRMLOAD\_FIRST

の の  
ffff,f□f,,□,,,,□ffff□fff,IDirect3DRMMeshBuilder::Load  
f,‘,ffffff,Š,,,,,,,,□‘,ffffff,Š,,,,,,,,□

## D3DRMLOAD\_INSTANCEBYREFERENCE

D3D,,,ffff,,□‘□,□□□□,,ffffff,□□,‘,,,□,,ffffff,f□□,,□  
ŠLoadffff,□lpvObjSourceffff□f,Ž–  
,,ffffff,f□f,Ž‘,□lpvObjIDffff□f,,,ffffff,Ž•,,□ffff,f□f□fff,‘‘,Š,,,lpv  
ObjSourceffff□f,%Ž,,□,,□Ž•Žfff,‘‘,Š,,,lpvObjIDffff□f,“–,%Ž,,□  
ffffff□fff,□fff□f,%Ž•–,,%,•□,%,,,□D3DRMLOAD\_INSTANCEBYREFERENCEfff,–  
,,□□,,ffffff,□□,,,,,,“,ffff,2%f□f,,,,,,□ffffff,–‘,Ž,,,,□□□D3DRMLO  
AD\_INSTANCEBYREFERENCEffff,fff,,,□D3DRMLOAD\_INSTANCEBYCOPYINGfff,fff,,  
,,“,“%,“,,,,,,,□f□f,□“^,ffffff,‘□,,,□Š–‘,ffffff,□,,ffffff,,□□,,□

xe "D3DRMMAPPING"\$  
D3DRMMAPPING

typedef DWORD D3DRMMAPPING, D3DRMMAPPINGFLAG;

static const D3DRMMAPPINGFLAG D3DRMMAP\_WRAPU = 1;

static const D3DRMMAPPINGFLAG D3DRMMAP\_WRAPV = 2;

static const D3DRMMAPPINGFLAG D3DRMMAP\_PERSPCORRECT = 4;

IDirect3DRMMesh::GetGroupMapping,IDirect3DRMMesh::SetGroupMapping

ing D3DRMMAPPINGFLAG

D3DRMMAPPING,“,

D3DRMMAP\_WRAPU

u

D3DRMMAP\_WRAPV

v f,,

D3DRMMAP\_PERSPCORRECT

ffff fffff,%o<• ,,,

D3DRMMAP\_WRAPU,D3DRMMAP\_WRAPVfff,□ffffff,ffffff□•,%Ž,,•  
–,“

の , u,,,v□•,—“’, fffff□fff,Ž,•,,,,^,,□^•,,,—

•,fff,fff,,,□‘□,“v•“•ffff,•,fff□ffffff,,□,,□□□ffffff,%o’ ,f□ff のの

IDirect3DRMWrap

Direct3DRMWrap Interface, IDirect3DRMMesh::GetGroupMapping,  
IDirect3DRMMesh::SetGroupMapping

```

xe "D3DRMMATRIX4D"§
D3DRMMATRIX4D
 typedef D3DVALUE D3DRMMATRIX4D[4][4];

 Š, —,,•, —ffff, D3DRMMATRIX4D[row][column],,,

typedef DWORD D3DRMSAVEOPTIONS;
#define D3DRMXOFSAVE_NORMALS 1
#define D3DRMXOFSAVE_TEXTURECOORDINATES 2
#define D3DRMXOFSAVE_MATERIALS 4
#define D3DRMXOFSAVE_TEXTURENAMES 8
#define D3DRMXOFSAVE_ALL 15
#define D3DRMXOFSAVE_TEMPLATES 16

IDirect3DRMMeshBuilder::Saveffff,ffffff,'<,,□
D3DRMXOFSAVE_NORMALS

D3DRMXOFSAVE_TEXTURECOORDINATES

D3DRMXOFSAVE_MATERIALS

D3DRMXOFSAVE_TEXTURENAMES
Š–“,-,%o, fffff,-‘,•‘,,
D3DRMXOFSAVE_ALL
Š–“,-,%o, – ffff fffff • fffff fffff,-‘,•‘,,
D3DRMXOFSAVE_TEMPLATES

Direct3D のCOM
D3DRM_OK

D3DRMERR_BADALLOC

```

D3DRMERR\_BADDEVICE

ffff,ffff ,ŒŠ ,,,

D3DRMERR\_BADFILE

f fffff,•  
D3DRMERR\_BADMAJORVERSION

DLL,fff f fff,•

D3DRMERR\_BADMINORVERSION

DLL,fff f fff,•

D3DRMERR\_BADOBJECT

^ „ffffff,•—

D3DRMERR\_BADTYPE

• ,Œ,^ ,“,,,  
D3DRMERR\_BADVALUE

• ,’,^ ,“,,,

D3DRMERR\_FACEUSED

–,ffff ,Ž— ,,,,

D3DRMERR\_FILENOTFOUND



*ffff*,Š,,,,,,,,,

D3DRMERR\_NOTDONEYET

Ž',,,,,,

D3DRMERR\_NOTFOUND

Ž',,,•–,ffffff,Œ,,,,,

D3DRMERR\_UNABLETOEXECUTE

*fff*□*ff*,Ž□,,,,□