

‘6

DirectInput

μDirectX 3, §, , , • §

DirectInput§— §

DirectInput,,ffffffffff,ff f §

DirectInput,,f f f,,fff,ff f §

DirectInputffffff §

fff §

§ §

f ffff§ §

IClassFactory..... §

IDirectInput..... §

IDirectInputDevice..... §

“..... §

ff ff• §

, §

f f f ffff’ §

fff ffff’ §

DirectInputffff fff< f f..... §

DirectInputf f ff fff fff..... §

•’..... §

DirectX 3, §, , , • XE "DirectX 3, §, , , •" §
 (E ,Microsoft® DirectX™, ffffffff,,,,,fff,,f f f"—
 ffff,ff f,, fff,,f f f,‘,,API, COM,fffffff,,fff ff f,—,
 DirectX, fff,,f f f f f,‘,,• Win32 API,, ‘,ffff,,
 Windows 95,‘, ,,DirectInput
 1.0,,,, ffff ffffffff ffff,,ffff,%o”,,,, ,,joyGetPosEx,,,,,API f
 ffff fff—
 fffff(joy.cpl) ,,ffff ffffffff ffff,ff f,%o”,,VJOYD.VXD,Š—
 ,,ffff fff,, ,,,,
 f□fff 1.0,ff□f^—□DirectX SDK,Š,□,%o—
 ,—,‘,,,”,,,, fff,,f□f□f□f□f,fffffff,,,,,Windows,,,,,□‘%o,,,,, ,,—
 f□fff,DirectInput,—•,,,,,□DirectInput
 3.0,,□fff,,f□f□f□f□f,‘,,,, ‘,ffff,%o”,, DirectInput 1.0
 ,,^,,□DirectInput 3.0 の API COM

DirectInput の COM
 ff□ffff,ff□f,,fffffff□ffff,‘,, DirectInput ,,ff□f,•%o,,□”—
 ffff(“,DirectInputfff□fff f,,,,,ffff),ff□f,‘,‘,•%o,,
 DirectInputŠ— XE "DirectInputŠ—" §

Windows DirectX SDK

,,,□□“,fffffff□ff□f,□□,,,,,,,□DirectInput,□“—
 ffff□f□f,,□‘,•—,ffff,%o”,,,DirectX
 SDKfff□fff,,,□

—□,—,□ff COM API COM DirectInput 1.0

DirectInput の XE "DirectInput の" §

XE " : " § XE " :
 ff " §DirectInput
 API,□ffff,,fff□fffffff,‘, ‘, ,,fff,‘<, ,,API Microsoft
 Win32® (SDK),fffffff API の ffff□f の
 ,, DirectInputfff□fff,□• ,,OEM<<,fffffff,□,^ ‘,fff の
 •,ffff,Š”,,
 ffffffff,ff□f,,DirectInputŠ□,ff
 Win32fffffffAPI,,□“,—, , DirectInput API,□fffffff□f,ffff

の DirectInput の

- の

- の
- の

の XE "の" §

DirectInput DirectInput

XE " :の " §DirectInput

のの

4 4 2

2 4 4 16 4 1

6 32

XE " : " §

ののの

XE " : " §

16 ののjoyGetNumDevs の 0

joyGetPosEx の JOYERR_NOERROR
JOYERR_UNPLUGGED

joyGetDevCaps

joyGetNumDevs の ‘,,,ffffffff ,,,, ffff,ff f,,ffffffff ,ž,

ffffffff,ffff fff,,fff XE "ffffffff,ffff fff,,fff" §

“”^,fff,, fffffffff,ffff fff XE "ffffffff:ffff fff"

§,,,fff,,,,, Microsoft Windows®

95,ffff ffff,,ffffffff ffff fff,, f f, ,,ffff fff,^%o

,ffffffff,“,,,

- ^”“,ffffffff XE "ffffffff:^”“,ffffffff"•EM XE "
- :OEM " §
- XE " : " §

の 6 32 ののののの

joyConfigChanged のの

の XE "の" §

joyGetPosEx の x y z Win32 API の 4 joyGetPosEx の

• 4 5 6 の r u v の

• XE " : " §

• XE " : " §

•••

• の の XE " : " §

DirectInput の XE "DirectInput の" §

の DirectInput の

• DirectInput

• DirectInputDevice

• の

•

•

•

DirectInput XE "DirectInput " §

DirectInput DirectInput
DirectInputCreate API DirectInput
DirectInputCreate API IDirectInput

IDirectInput の DirectInput
IDirectInput::EnumDevices

DirectInputDevice XE "DirectInputDevice " §

—ffff, DirectInputDevice の

□•,

□ffff□fff, IDirectInput::CreateDeviceffff, CE

, □,, DirectInputDeviceffff, □□,, □

IDirectInput::CreateDeviceffff, IDirectInputDeviceffff□ff □□,•, □

IDirectInput::CreateDevice, ^, fff□f, fffffG,, GUID
GUID_SysMouse GUID_SysKeyboard の " §

XE " : " § c_dfDIMouse
IDirectInputDevice::SetDataFormat

,<fff,ff f,,,, DISCL_BACKGROUND |
DISCL_NONEXCLUSIVE, DirectInput 3.0 DISCL_BACKGROUND |
DISCL_EXCLUSIVE のの'€' ,,,,< ,Ž ,,,,
DirectX, —f fff,, Windows NT ,DirectInput,ff f,,,,, ,, Windows
NT,,fff ffff,',,DISCL_FOREGROUND |
DISCL_EXCLUSIVE,,ff f,,, fff ffff,ff ffff,,,,, Windows
95 ,DirectInput,,DISCL_FOREGROUND |
DISCL_EXCLUSIVE,,ff f,€',,,%o" ,, ,,,, ,,,,,,ffffff f,,,fff
ffff ,ffff fff,Ž ,•—,,, ,,,,<fff,—,,,,, ,,Windows NT%o
,DINPUT.DLLffff f f fff(DirectX 3.0 SDK,Extrasffffff,,),—
,,fff,•—,,, Windows
NTf fff,DirectInput(,,fff ffff,ff f,,,,,Windows
95f fff,DirectInput),Windows
95f fff,,,"DIERR_INPUTLOST,",,,, ,,,f f, ',,,,,ffff fff, ,,,,
,,,"

XE "ffff,"Ž:fff□ffff"
§IDirectInputDevice::GetDeviceData,,IDirectInputDevice::GetDeviceState,,ff
f□ffff,,,,,,,□ffff□fff,",,,“—
ff□ff,Ž,,,,,□IDirectInputDevice::Unacquire,€,□,,ffff□fff,,fff□ffff
,%o*,,,,□□,,□ffff□fff□fff□,,,ffff□fff□," ,,,, ,
fff,fff,,," , ffff,%o*,,,,,,□ffff□f ,",%o ,,,,ffff□fff,“—
ff□ff,□“Š“,,,,, ffff□ f,IDirectInputDevice::Acquire
ff ,',•—,,

ffff□fff,DISCL_FOREGROUND —
,,, IDirectInputDevice::GetDeviceData,, IDirectInputDevice::GetDeviceSt
ate,,DIERR_INPUTLOST',fff,,,,□□,,□ffff□fff,ff□ff,Ž,□□DirectI
nput,Ž““ ,fff,%o*,,,,□DIERR_INPUTLOST ,Ž,Ž,,□□
ff□fff,fff□ffff,"Ž,□“ ,,,, ,,, ,,,,fff,ff□," ,,,,,—
€f□f,Š,%o" ,,,,,,

ffff□fff,DISCL_NONEXCLUSIVEf□f, f,fff,,,
f,DirectInput€—,Windowsfff□fff□f€—,—
•,Ž“ ,,, ffff□fff,DISCL_EXCLUSIVEf□f,fff,fff,,□□fff□f□f,D
irectInput ,,,Ž“%o” ,,, Windowsfff□fff□f,Ž—,,f□f,Š,,,□

Scrawlffff ff□fff,□fff□ffff

□□□□□□□□□□□□□□□□ XE

"f□f□f□ffff,,f□fŽ“" §

XE "f□f□fe: DirectInput
ののの
PC の DIK_ dinput.h
NEC PC のDirectInput NEC
PC の

XE "の: " §の256 のの IDirectInputDevice::GetDeviceState
IDirectInputDevice::GetDeviceState Windows の
GetKeyboardState ののの 256 ののの DIK_ENTER ののEnter
GetKeyboardState DirectInput のの

XE " の : " §
 IDirectInputDevice::GetDeviceData
 DirectInput
IDirectInputDevice::GetDeviceData

の DIDEVICEOBJECTDATA の
 の
 "ÒA"
 IDirectInputDevice::GetDeviceData の DIDEVICEOBJECTDATA の 3
 の "A"
 "ÒR"
 IDirectInputDevice::GetDeviceData のの
 DIK_* DIDEVICEOBJECTDATA の dwOfs の
 () DIK_*
 DIDEVICEOBJECTDATA のののの dwData のの dwData の
 IDirectInputDevice::GetDeviceData の

XE " : " § ; の
IDirectInputDevice::SetCooperativeLevel Windows 95Š‰%
 ,, f□f□f□ffff,‘,,‰%<’fff,ff□f,,,,,□ DISCL_BACKGROUND |
 DISCL_NONEXCLUSIVE DISCL_FOREGROUND |
 DISCL_NONEXCLUSIVE DirectInput 3.0
DISCL_BACKGROUND | DISCL_EXCLUSIVE DISCL_FOREGROUND
 | **DISCL_EXCLUSIVE** f,f□f□f□ffff,‘,,ff□f,,,,,
 f□f□f□f,, DirectInput Windowsf ; ^
 Š<’fff –,Š,,□のの

DirectX,□—f□fff,,□Windows NT DirectInput,ff□f,,,,,,□,, Windows
 NT f□f□ffff,‘,,DISCL_FOREGROUND | DISCL_NONEXCLUSIVE
 Windows NT

の の
Windows 95 の DirectInput DISCL_FOREGROUND |
DISCL_NONEXCLUSIVE の

XE "の: f□f□ "
 §IDirectInputDevice::GetDeviceData,,IDirectInputDevice::GetDeviceState
IDirectInputDevice::Acquire

IDirectInputDevice::Unacquire

IDirectInputDevice::Acquire

DISCL_FOREGROUND
 IDirectInputDevice::GetDeviceData IDirectInputDevice::GetDeviceState
 の DIERR_INPUTLOST
 DirectInput
 DIERR_INPUTLOST
 DirectInput

XE " " §
 XE " : " §
 の §,“,,, ’,,,,,fff□fff,□□, P US PC
 の PC XE " :Windows " §Windows
 (DIK_LWIN DIK_RWIN DIK_APPSの
 XE " : " §DIK_PAUSE
 PC DIK_PAUSE のDIK_LCONTROL
 DIK_NUMLOCK の òPause”
 XE " :の" § のの XE " :
 の" §

の XE " :PC XT " §PC XT PC
 ATK_F11 DIK_F12 の(DIK_* 0x80)の PC XT
 DIK_SYSRQ

NEC PC-98 US

XE "

US

NEC PC-98 XE " :NEC PC-98 " §
 DIK_CAPSLOCK DIK_KANA ,,

DOS/V 106 ,
 NEC PC-9ONVERT
 DIK_KANA
 DIK_KANJI
 DIK_NOCONVERT
 DIK_YEN
 DIK_APOSTROPHE
 DIK_EQUALS
 DIK_GRAVE

NEC PC-98

NEC PC-98
DIK_AT
DIK_CIRCUMFLEX
DIK_COLON
DIK_F13, F14, F15
DIK_KANA
DIK_KANJI
DIK_NOCONVERT
DIK_NUMPADCOMMA
DIK_NUMPADEQUALS
DIK_STOP
DIK_UNDERLS
DIK_GRAVE
DIK_NUMLOCK
DIK_NUMPADENTAX
DIK_AX
DIK_CONVERT
DIK_KANJI
DIK_NOCONVERT
DIK_YEN
DIK_RCONTROL
DIK_RMENU

J-3100
DIK_KANA
DIK_KANJI
DIK_NOLABEL
DIK_YEN
DIK_RCONTROL
DIK_RMENU

DirectInput XE "DirectInput" §

DIDFT_GETINSTANCE XE "DIDFT_GETINSTANCE" §

BYTE DIDFT_GETINSTANCE(
 DWORD dwType)

DirectInputf□f□ff□fff□ §

BYTE DIDFT_GETTYPE(
 DWORD dwType)

DirectInput

dwType

DirectInput

DISEQUENCE_COMPARE XE "DISEQUENCE_COMPARE" §

BOOL DISEQUENCE_COMPARE(
 DWORD dwSequence1, cmp,
 DWORD dwSequence2)

① DirectInput =" "<" ">" "<=" ">="

dwSequence2

①

①0

① dwSequence1 dwSequence2

if (DISEQUENCE_COMPARE(dwSequence1, <, dwSequence2)) {

```

...
}

① dwSequence1  dwSe

```

```

quence2

if
(DISEQUENCE_COMPARE(dw
Sequence1, >=, dwSequence2)) {
...
}

GET_DIDEVICE_SUBTYPE XE
"GET_DIDEVICE_SUBTYPE" §
BYTE
GET_DIDEVICE_SUBTYPE(
    DWORD dwDevType)

ののの DirectInput

dwDevType
DirectInput

GET_DIDEVICE_TYPE XE
"GET_DIDEVICE_TYPE" §
BYTE GET_DIDEVICE_TYPE(
    DWORD dwDevType)

DirectInput

dwDevType

```

--	--

<p>DirectInput</p> <p>MAKEDIPROP XE "MAKEDIPROP" §</p> <p>Microsoft</p> <p>GUID</p> <p>DirectInputCreate XE "DirectInputCreate" §</p> <p>HRESULT DirectInputCreate(HINSTANCE hinst, DWORD dwVersion, LPDIRECTINPUT * lplpDirectInput, LPUNKNOWN punkOuter);</p> <p>IDirectInput COM DirectInput</p> <p>*lplpDirectInput</p> <p>⓪ punkOuter = NULL</p> <p>CoCreateInstance(&CLSID_DirectInput, punkOuter, CLSCTX_INPROC_SERVER, &IID_IDirectInput, lplpDirectInput) Initialize</p> <p>⓪ punkOuter != NULL</p>		
---	--	--

<p>CoCreateInstance(&CLSID_DirectInput, punkOuter, CLSCTX_INPROC_SERVER, &IID_IUnknown, lplpDirectInput)</p> <p>の AN</p>		
--	--	--

Sif□fff,,UNICODEf□fff,*,□,,□ANSIf□fff, IDirectInputWfff□fff,ff□f,,ffffff,□□,,
 □create,□,*,žfff,ffffff,,□
 hinst
 DirectInputffffff,□□,,,fff□fff,,DLL,ffffff ffff□
 dwVersion

dinput.h の DIRECTINPUT_VERSION

DirectInput の の DirectInput
 DLL の

lplpDirectInput
 IDirectInputfff ffff,,fff,,*,*,ž,ž,

punkOuter
 OLE□□%o<”,-’, Ć,,fff□fff□fff,□□%o
 ,,irectInput,*,,,ffffff,IUnknown,,fff,,□
 OLE□□’,—<, IDirectInput,,fff,,,,

COM の

DI_OK = S_OK:

DIERR_INVALIDPARAM = E_INVALIDARG: lplpDirectInput

DIERR_OUTOFMEMORY = E_OUTOFMEMORY: fff,*,,,

DIERR_DIERR_OLDDIRECTINPUTVERSION:
 DirectInput,f□ffctInputf□f□f□fff’,,fff□fff,□,,,□

joyConfigChanged XE "joyConfigChanged" S
 MMRESULT joyConfigChanged(
 DWORD dwFlags);

ffff,□,,ffffff,□’

のの

の ‘,,,,, ffff ffff,,,ffffff
fff,ffffff,,E, ,,

OEM,ffffff□ffff□fff□ffff□fff,,,□ffff□fff,ffffff,fff□f
ff,ffffff,,,□
ff,‘,ffff,JOYSTICK USER’,•Q,,

JOYSTICK USER の

HKEY_LOCAL_MACHINE

dwFlags

0

•’

JOYERR_NOERROR の

JOYERR_NOCANDO

JOYERR_REGISTRYNOTVALID

```
xe "joyGetDevCaps"§
joyGetDevCaps XE "joyGetDevCaps" §
MMRESULT joyGetDevCaps(
    UINT uJoyID,
    LPJOYCAPS pjc,
    UINT cbjc);
```

ffff,Ž,ć”,fff

の

uJoyID

のJOYSTICKID1 □□ JOYSTICKID2,,,□

pjc

ffffffć”,Š,JOYCAPS□“,fff□

cbjc

JOYCAPS□“,fff,fff□,Ž,□

•’

□E,,, JOYERR_NOERROR □Ž”,,,Ž,ff□,*,□

MMSYSERR_INVALIDPARAM

MMSYSERR_NODRIVER

```

JOYERR_REGISTvs"$
joyGetNumDevs XE "joyGetNumDevs" $
UINT joyGetNumDevs(VOID);

```

fffffff□*ffff,ff*□*f,,,**ffff*

joyGetPosEx

0

joyGetDevCaps

xe "joyGetPosEx"\$

joyGetPosEx XE "joyGetPosEx" \$

```

MMRESULT joyGetPosEx(UINT uJoyID, LPJOYINFOEX pji);

```

^,,fff ‘,fffffff,,Ž“,,

„Š □□ €□;□JOYINFOEX□“,dwFlags—‘,„,^fff□Ž“fff□‘□,fff,Ž,ffff,„Š'ffff,„□'6Ž,—

„□•€„fff,%“„,□

uJoyID

□%„,fffffff,Ž•Ž□

pji

○ JOYINFOEX ○

JOYERR_NOERROR ○

JOYERR_UNPLUGGED

MMSYSERR_BADDEVICEID

MMSYSERR_INVALIDPARAM

MMSYSERR_NODRIVER

Ž JOYINFOEX

f ffffŠ

DIEnumDeviceObjectsProc XE ACK DIEnumDeviceObjectsProc(
LPCDIDeviceObjectInstance lpddoi,
LPVOID pvRef);

IDirectInputDevice::EnumObjects

lpddoi

□□□□□□□□□□□□□□□□□□

DIDeviceObjectInstance □ “,ffff□

pvRef

IDirectInput“,□

•’

DIENUM_CONTINUE —<Æ’,

DIENUM_STOP —<Ž’,

DIEnumDevicesProc XE "DIEnumDevicesProc" S

BOOL CALLBACK DIEnumDevicesProc(
LPDIDeviceInstance lpddi,

LPVOID pvRef

);

IDirectInput::EnumDevicesffff,—,f□ffffŠ□□

lpddi

—<,,,,ffff,< ,DIDeviceInstance の

pvRef

IDirectInput::EnumDevices

∞∞

DIENUM_CONTINUE

DIENUM_STOP

IClassFactory XE "IClassFactory" §

IClassFactory OLE
 OLE DirectInput の
 OLE

IClassFactory::AddRef XE "IClassFactory\:\:AddRef" §

HRESULT AddRef(
 LPCLASSFACTORY *lpClassFactory*);

OLE の

IUnknown::AddRef

IClassFactory::CreateInstance XE "IClassFactory\:\:CreateInstance" §

HRESULT CreateInstance(
 LPCLASSFACTORY *lpClassFactory*,
 LPUNKNOWN *punkOuter*,
 REFIID *riid*,
 LPVOID * **ppvOut**);

DirectInput

OLE の

IClassFactory::CreateInstance

punkOuter

OLE の
 の

riid

ppvOut

の

COM の

S_OK:

E_INVALIDARG: ppvOut

CLASS_E_NOAGGREGATION:

E_OUTOFMEMORY:

E_NOINTERFACE:

IClassFactory::LockServer XE "IClassFactory\:\:LockServer" §

HRESULT LockServer(

LPCLASSFACTORY lpClassFactory,

BOOL fLock);

DLL

0

OLE

DLL

0

IClassFactory::LockServer

fLock

TRUE fff ffff,ffffff,,

FALSE,,, fff ffff,ffffff,,

•'

COMff f f,•, %0,ff f f,•,,f f,,,,,E,,

S_OK:

E_OUTOFMEMORY: fff,•',,,,

IClassFactory::QueryInterface XE "IClassFactory\:\:QueryInterface" §

HRESULT QueryInterface(

LPCLASSFACTORY lpClassFactory,

REFIID riid,

LPVOID * ppvObj);

ffffff ;',fff ffff,ffffff ffff,—

,, OLEffffff,IUnknown::QueryInterface,Ž ,,,,

riid

•—,

IID

ppvObj

,,ffff,Ž,Ž,

•'

COMff f f,•,

IClassFactory::Release XE "IClassFactory\:\:Release" §
HRESULT Release(LPCLASSFACTORY lpClassFactory);

fff ffff,Ž ffff,ffffff,, ffffff,Ž ffff,0,, ,ffffff,fff,,
 %o•,, OLEfffff,IUnknown::Release,Ž ,,,
 •'
 ffffffŽ ffff,•,

IDirectInput XE "IDirectInput" §
DirectInputffffff,DirectInput**ffffff**,•, "—
 ffff,•,DirectInputDeviceffffff, ,,

IDirectInput::AddRef XE "IDirectInput\:\:AddRef" §

DWORD AddRef(

LPDIRECTINPUT lpDirectInput);

IDirectInput	IUnknown の COM	
AddRef		1 Release
		0
の		
lpDirectInput		
の DirectInput		

DWORD

IDirectInput::CreateDevice XE "IDirectInput\:\:CreateDevice" §
HRESULT CreateDevice(

	LPDIRECTINPUT
lpDirectInput,	
	REFGUID rguid,
	LPDIRECTINPUTDEVICE
*lplpDirectInputDevice,	
	LPUNKNOWN * pUnkOuter);

GUID

の **punkOuter** = NULL
 CoCreateInstance(&CLSID_DirectInputDevice, NULL,
 CLSCTX_INPROC_SERVER, riid, lplpDirectInputDevice),%
 ,,ffffff,□□,Init,,□ CoCreateInstance(&CLSID_DirectInputDevice,
 punkOuter, CLSCTX_INPROC_SERVER, &IID_IDirectInput, lplpDirectInput)

lpDirectInput

の DirectInput

rguid

GUID の GUID EnumDevices
GUID の

lpDirectInputDevice

IDirectInputDevice

punkOuter

OLE の
0 の

0

COM の

DI_OK = S_OK:

DIERR_INVALIDPARAM = E_INVALIDARG: ppvOut

DIERR_OUTOFMEMORY = E_OUTOFMEMORY:

DIERR_NOINTERFACE = E_NOINTERFACE: の

DIERR_DEVICENOTREG: DirectInput

IDirectInput::EnumDevices XE "IDirectInput\:\EnumDevices" §

HRESULT EnumDevices(

LPDIRECTINPUT lpDirectInput,

DWORD dwDevType,

LPDIENUMCALLBACK

lpCallback,

LPVOID pvRef,
DWORD dwFlags);

‘,

”

←TACHEDONLY,fff,,,,,□fff,,,,□

IDirectInput,CE ,,IUnknownfff ffff,^• ffff fff,Ž—
 ,,,,,fff ffff,,fffff,ff f,,,,,—, ,,,—, —
 , ,,“,COMfff ffff,fffff,ff f,,, ffff fff,,fff ffff,’,Ž
 —,Ž, ppvObj の
 Release ,,fff ffff,%o•,,,,,, ,,fff,, Š‘,,, —
 ,<”, “ Š ,,,,,DirectInputfffff,Š’,,, ,,,, OLEfffff,IUnknown:
 :QueryInterface,Ž ,,,

lpDirectInput

,,fff,CE, ,,DirectInputfffff,Ž,Ž,
 riid
 •—,fff ffff,Ž•,fff ffffŽ•Ž(IID),Ž,Ž,
 ppvObj
 -,□,,,□□,Ž□,,,□□•,,,fff□ff

- DI_OK
- DIERR_INVALIDPARAM
- DIERR_NOINTERFACE

**IDirectInput::Release XE "IDirectInput\:
 \:Release" §**

DtInput,CE□,,IUnknownfff□ffff,^□Š~,,,COMfffff,Ž□fff,CE,,,—
 ,,□fffff,□□,□□,,,□Ž□fff,1,fff,,□AddRef,CE,□,,,Ž□fff,fff
 fff,,□Release,CE,□,,,Ž□fff,ff 0
 の OLE の

lpDirectInput

fff,CE, ,,DirectInputfffff,Ž,Ž,
 •
 ,,Ž ffff,Š,DWORD ,,’,fff—“,,Ž—,,,,,’^

IDirectInput::RunControlPanel XE "IDirectInput\:\:RunControlPanel" §
 HRESULT RunControlPanel(

LPDIRECTINPUT

lpDirectInput,

**HWND hwndOwner,
 DWORD dwFlags);**

Windows の DirectInputfff□ffff,Ž□,,,—,,,□f□□ **DirectInput**

hwndOwner


```
DI_BUFFEROVERFLOW = S_FALSE:
f□f,□□,Ž" ,,,□ffff.ffff□fff,□• ,,,,,,f□f ,,,,Ž ,,,□,□□□ffff□f□f,Ž"%□' ,,,a,□□,CE,□ ,,,,• ,,,
□, ,f□f,□□ff□ff□f□f ,,,,^□
```

```
DIERR_NOTACQUIRED: ffff,Ž ,,,,□
```

```
DIERR_INPUTLOST: ffff,ffff, ,,,□ffff□fff,ffff,□Ž ,,,,□
```

```
DIERR_INVALIDPARAM = E_INVALIDARG: -CE,ff
```

の10

```
DIDeviceObjectData rgdod[10];
```

```
DWORD dwItems = 10;
```

```
hres = IDirectInputDevice_GetDeviceData(
    pdid,
```

```
    sizeof(DIDeviceObjectData),
```

```
    rgdod,
```

```
    &dwItems,
```

```
    0);
```

```
if (SUCCEEDED(hres)) {
```

```
    // ffff,□□,fffff,
```

```
    // dwItems =
```

```
    if (hres == DI_BUFFEROVERFLOW) {
```

```
        // ffff,f□-□,• ,□
```

```
        dwItems = INFINITE;
```

```
        hres = IDirectInputDevice_GetDeviceData(
```

```
            pdid,
```

```
            sizeof(DIDeviceObjectData),
```

```
            NULL,
```

```
            &dwItems,
```

```
            0);
```

```
if (SUCCEEDED(hres)) {
```

```
    //
```

```
    // dwItems =
```

```
    if (hres == DI_BUFFEROVERFLOW) {
```

```
        //
```

```

    }
}

```

```

rgdod NULL
ff,,f

```

```

dwItems = INFIata(
    pdid,
    sizeof(DIDEVICEOBJECTDATA),
    NULL,
    &dwItems,
    DIGDD_PEEK);
if (SUCCEEDED(hres)) {
    // dwItems =
    if (hres == DI_BUFFEROVERFLOW) {
        // ffff,f ff,
        // ,,f,ff,-,,""
        dwItems = 0;
        hres = IDirectInputDevice_GetDeviceData(
            pdid,
            sizeof(DIDEVICEOBJECTDATA),
            NULL,
            &dwItems,
            0);
    }
    if (hres == DI_BUFFEROVERFLOW) {
        //
    }
}

```

```

IDirectInputDevice::GetDeviceInfo XE "IDirectInputDevice\:\:GetDeviceInfo" §
HRESULT GetDeviceInfo(
    LPDIRECTINPUTDEVICE lpDirectInputDevice,

```


DIERR_NOTACQUIRED:

DIERR_INPUTLOST: \emptyset

DIERR_INVALIDPARAM = E_INVALIDARG: lpvData \emptyset SetDataFormat
cbData

IDirectInputDevice::GetObjectInfo XE "IDirectInputDevice\:\:\GetObjectInfo" §

HRESULT GetObjectInfo(

LPDIRECTINPUTDEVICE lpDirectInputDevice,

LPDIDEVICEOBJECTINSTANCE pdidoi,

DWORD dwObj,

DWORD dwHow)

lpDirectInputDevice

,,fff,☒, ,,DirectInputffff fffffff,Ž,Ž,

pdidoi

ffffff,Š,, •,Ž,Ž, , \emptyset

☒, ‘,DIDEVICEOBJECTINSTANCE “,dwSizeff ff, Š%o,,,,,,,,,

dwObj

ffff,ffff,,ffffff,Ž•,,

dwHow

dwObj,%oŽ•-,Ž•,,

Ž DIPROPHEADER

•’

COMff□f□f,•,□^%o,ff□f□f,—□“,,,^□ \emptyset

DI_OK = S_OK: , —,,

DIERR_INVALIDPARAM = E_INVALIDARG: -☒,fff f,‘ ,,

DIERR_OBJECTNOTFOUND: Ž’,,ffffff,‘ ,,

IDirectInputDevice::GetProperty XE "IDirectInputDevice\:\:\GetProperty" §

HRESULT GetProperty(

```
LPDIRECTINPUTDEVICE lpDirectInputDevice,  
REFGUID rguidProp,  
LPDIPROPHEADER pdiph);
```

```
IDirectInputDevice::SetProperty  
IDirectInputDevice::SetProperty
```

```
lpDirectInputDevice
```

```
の DirectInput  
rguidProp
```

```
DIPROP_*の GUID の
```

```
(C++ )  
の
```

```
DIPROP_AXISMODE  
DIPROP_BUFFERSIZE  
DIPROP_GRANULARITY  
DIPROP_RANGE
```

```
ののののの  
pdiph
```

```
DIPROPHEADER
```

```
COM のの
```

```
DI_OK = S_OK;
```

```
DIERR_INVALIDPARAM = E_INVALIDARG: pdiphfff□f,-□E,ffff,,,,□dwHowfff□ff,-  
□E,□,,,,fffff;□,,,□
```

```
DIERR_UNSUPPORTED = E_NOTIMPL:ffff,,,fffff,fffff,ff□f,,,,□
```

```
—  
^%,C□E□f□f,^*,□DIPROP_BUFFERSIZEfffff;,'Ž“•-,Ž,,,,□
```

```
DIPROPDWORD dipdw;
```

```
HRESULT hres;
```

```
dipdw.diph.dwSize = sizeof(DIPROPDWORD);
```

```
dipdw.diph.dwHeaderSize = sizeof(DIPROPHEADER);
```

```
dipdw.diph.dwObj = 0; // の
```

```
dipdw.diph.dwHow = DIPH_DEVICE;
```

```
hres = IDirectInputDevice_GetProperty(pdid, DIPROP_BUFFERSIZE,  
&dipdw.diph);
```

```
if (SUCCEEDED(hres)) {
```

```
// dipdw.dwData
```

```
}
```

```
IDirectInputDevice::EnumObjects XE "IDirectInputDevice\:\EnumObjects" §
```

```
HRESULT EnumObjects(  
LPDIRECTINPUTDEVICE lpDirectInputDevice,  
LPDIENUMDEVICEOBJECTSCALLBACK lpCallback,
```

```
LPVOID pvRef,
DWORD fl);
```

```
( )
```

```
lpDirectInputDevice
```

```
の DirectInput
```

```
lpCallback
```

```
DirectInputDevice
```

```
BOOL CALLBACK DIEnumDeviceObjectsProc(  
LPCDIDEVICEOBJECTINSTANCE lpddoi,  
LPVOID pvRef);
```

```
pvRef
```

```
( )
```

```
fl
```

```
DirectInput
```

```
COM の
```

```
DI_OK = S_OK:
```

```
DIERR_INVALIDPARAM = E_INVALIDARG: fl
```

```
IDirectInputDevice::Initialize XE "IDirectInputDevice\:\:\Initialize" §
```

```
HRESULT Initialize(
```

```
LPDIRECTINPUTDEVICE lpDirectInputDevice,
```

```
HINSTANCE hinst,
```

```
DWORD dwVersion,
```

```
REFGUID rguid);
```

DirectInputDevice

の
,,, “ Š%o,•—,„„„,^ „„„„ fffffff, ‘,’,„„„,Œ,„„, ‘,„„„,ffffff,Ž—
IDirectInput::CreateDeviceffff,ffff, „„,Ž““ „„„ Š%o
„„„ ’ ,ffff fff,„„,Š ,Œ, ,•—,„„

lpDirectInputDevice

„,fff,Œ, „„,DirectInputffff fffffff,Ž,Ž,
hinst
DirectInputDeviceffffff, „„„,ffff fff,„„,DLL,ffffff ffff DirectIn
put, „,’,—,ffff fff,„„,DLL,„ „„„„„„,’,

dwVersion

Ž—,„„,diinput.h のの
,DIRECTINPUT_VERSION,„„„„„„,„

DirectInput,„,’,—

„, ,f fff,DirectInput,‘,ffff fff,„„,DLL, Œ,„„„„„,Š’,„

rguid

fff□ffff,Š~„„,ffff,ffffff,Ž•,„□IDirectInput::Enum
Devicesfff,—
„,□ffff,„,ffffffGUID,ff□f,„„„„,’,„„„„„□
•
COMff□f□f,•,□^%o,ff□f□f,•,„„,f□f,„„„„,Œ,„„□
DI_OK = S_OK: fff, ‘,„„„

DIERR_DIERR_OLDDIRECTINPUTVERSION: DirectInput の

DIERR_DIERR_BETADIRECTINPUTVERSION:

ff□fŠ„„,DirectInputf□f□f□fff,‘,„,ffff□f
S_FALSE: ffff,rguid,“„„,ffffffGUID,„„„ Š%o,„„„
DIERR_ACQUIRED: ffff,Ž„„„„, Š%o,„„„„

IDirectInputDevice::QueryInterface XE "IDirectInputDevice\
\:QueryInterface" §

HRESULT QueryInterface(

LPDIRECTINPUTDEVICE lpDirectInputDevice,

REFIID riid,

LPVOID FAR* ppvObj);


```

IDirectInputDevice::SetCooperativeLevel XE "IDirectInputDevice\:\SetCooperativeLevel" §
HRESULT SetCooperativeLevel(
LPDIRECTINPUTDEVICE lpDirectInputDevice,

HWND hwnd,

DWORD dwFlags);

```

のののののの

```

Acquire                               の,,,,,
hwnd
ffff,Š~,fffff
DISCL_FOREGROUNDfff,“,,,,,   ,,fff f,NULL~Š,,,,,,, ,,fffff,
fffff,,,,,,, DirectInput   ,fffff,,Š,fffff,•,,ff ,,
dwFlags
ffff,Š~,‘fff,‹ ,,fff
DISCL_*fff,,,,,•“• %o,,,,,
•’
COMff f f,•, ^%o,ff f f,•,,f f,,,,,E,,,
DI_OK = S_OK: ‘ , , —,,
DIERR_INVALIDPARAM = E_INVALIDARG: hwndfff f,-
E,fffff ffff,,,,-E,fff,,fff,‘, ,,“,,,

```

```

IDirectInputDevice::SetDataFormat XE "IDirectInputDevice\:\SetDataFormat" §
HRESULT SetDataFormat(
LPDIRECTINPUTDEVICE
lpDirectInputDevice,
LPCDIDATAFORMAT lpdf);

```



```
    ,,, DirectInput,ffff'ffff,“,,,,ResetEvent,☉, ,,
    ,ffff fff,,ffff,Ž“ffff ffff,, ,,
```

```
hEvent,NULL,    ’,%o”,,,
•’
```

```
COMff f f,•    ☉    ,,f f,,,,,☉,,,
DI_OK = S_OK: ‘ , , —,,
```

```
DIERR_ACQUIRED:
IDirectInputDevicefffff,Ž,,,,☐IDirectInputDevice::Unacquire
☉
```

DIERR_HANDLEEXISTS: IDirectInputDevicefffff,☐ffff'ffff,,,,‘☐,,☐ DirectInp

```
E_INVALIDARG: ffff☐ffff,,,☐
—
ffff,☉Ž“,fff,,,,,☐ffffff,ffff,,☐
```

```
dwResult = WaitForSingleObject(hEvent, 0);
```

```
if (dwResult == WAIT_OBJECT_0) {
```

```
    //
    //
    //
```

```
}
```

```
☉☉( Windows )
```

```
dwResult = WaitForSingleObject(hEvent, INFINITE);
```

```
if (dwResult == WAIT_OBJECT_0) {
```

```
    //
    //
    //
```

```
}
```

```
HANDLE ah[2] = { hEvent1, hEvent2 };
```

```
while (TRUE) {
```

```
    dwResult = MsgWaitForMultipleObjects(2, ah, FALSE,
        INFINITE, QS_ALLINPUT);
```

```
    switch (dwResult) {
```

```
        case WAIT_OBJECT_0:
```

```
            // ffff1
```

```
            //
```

```
            //
```

```
            ProcessInputEvent1();
```

```
            break;
```

```
        case WAIT_OBJECT_0 + 1:
```

```
            // ffff2,fff,,
```

```
            // ffff,ž"ffffT_0 + 2:
```

```
            // Windowsffff□f,ž,ž,
```

```
            // ffff□f,,,,,□—□,
```

```
            while(PeekMessage(&msg, NULL, 0, 0, PM_REMOVE)){
```

```
                if (msg.message == WM_QUIT) {
```

```
                    goto exitapp;
```

```
                }
```

```
                TranslateMessage(&msg);
```

```
                DispatchMessage(&msg);
```

```
            }
```

```
            break;
```

```
        default:
```

```
            //
```

```
            Panic();
```

```
            break;
```

```
    }
```

```
}
```

```
HANDLE ah[2] = { hEvent1, hEvent2 };
```

```
DWORD dwWait = 0;
```

```
while (TRUE) {
```

```
    dwResult = MsgWaitForMultipleObjects(2, ah, FALSE,  
dwWait, QS_ALLINPUT);
```

```
    dwWait = 0;
```

```
    switch (dwResult) {
```

```
    case WAIT_OBJECT_0:
```

```
        // ffff1,fff,,
```

```
        // ffff,Ž“ffff,, ,,,
```

```
        // ,,,,ffff,,
```

```
        ProcessInputEvent1();
```

```
        break;
```

```
    case WAIT_OBJECT_0 + 1:
```

```
        // ffff2,fff,,
```

```
        // ffff,Ž“ffff,, ,,,
```

```
        // ,,,,ffff,,
```

```
        ProcessInputEvent2();
```

```
        break;
```

```
    case WAIT_OBJECT_0 + 2:
```

```
        // Windows
```

```
        //
```

```
    while(PeekMessage(&msg, NULL, 0,  
0, PM_REMOVE)){
```

```
        if (msg.message == WM_g);
```

```
        DispatchMessage(&msg);
```

```
    }
```



```

    numObjects,        //
    rgodf,             //
};

```

DIDEVCAPS XE "DIDEVCAPS" §

```
typedef struct {
```

```

    DWORD dwSize;

    DWORD dwDevType;

    DWORD dwFlags;
    DWORD dwAxes;
    DWORD dwButtons;
    DWORD dwPOVs;
} DIDEVCAPS;

```

IDirectInputDevice::GetCapabilities 0

•,

dwSize

□“,fff,fff,Ž’,, IDirectInputDevice::GetCapabilities,0,□,‘, Š%0

dwDevType

ffff fffŽ’Ž’,,ff ff,‘ ,,,, DirectInputffff fff< f f,Ž’, ,,,

dwFlags

ffff,Š’,,fff ~%0,fff,‘, ,,,,

DIDC_ATTACHED: ffff,•—“, ‘ ,,,,

DIDC_POLLEDDEVICE:

ffff,Š’,, <“0, ,,,f fff0, ,, f f,“ ,,,,ffff fff,-

Ž’,,GetDeviceState,0, ,,,, ,,, fff ,

dwAxes

ffff ,—%0”,Ž’, ,Ž’, ,

dwButtons

ffff ,—%0”,fff, ,Ž’, ,

dwPOVs

ffff ,—%0”,Ž’, f, ,Ž’, , DirectInputf fff3.0, ,— ,,,,

DIDEVICEINSTANCE XE "DIDEVICEINSTANCE" §

```
typedef struct {
```

```

    DWORD dwSize;

```

```

GUID guidInstance;
GUID guidProduct;
DWORD dwDevType;
TCHAR tszInstanceName[MAX_PATH];
TCHAR tszProductName[MAX_PATH];
} DIDEVICEINSTANCE;

```

IDirectInput::EnumDevices IDirectInputDevice::GetDeviceInfo

dwSize
 の
 guidInstance

GUID GUID
 ,, GUID の
GUID
guidProduct

dwDevType
 DirectInput
 tszProductName[MAX_PATH]
 の " Frobozz Industries SuperStick 5X"
 tszInstanceName[MAX_PATH]
 の Joystick 1"

```

DIDEVICEOBJECTDATA XE "DIDEVICEOBJECTDATA" §
typedef struct {
    DWORD dwOfs;
    DWORD dwData;
    DWORD dwTimeStamp;
    DWORD dwSequence;
} DIDEVICEOBJECTDATA;

```

IDirectInputDevice::GetDeviceData の

dwOfs

IDirectInputDevice::GetDeviceState dwData

,, **dwOfs** の
DIMOFS_* の
 DIK_*',,,,
 fffff f ff fff,fff,,,,, fffff f ff fff,'',''fffff
 dwData
 ffff,",,f□f□,,f□f,fff,fff,fff,^,□,□□,,f□f,,,,,EŽ,",,□
 DIDFT_AXIS: ffff,"Žf□f,,□□□"Ž%o,",,□ffff,□Žf□f,,□□□□Ž□,",,□
 DIDFT_BUTTON: dwData,%o^fff,,^,Ž,□fff,%o,,,%o^fff,□fff,fff,,□fff,—,,,fff,,□
 dwTimeStamp
 ff*^,";ffff□ffff,□,,,□,fff,□□,,,□E□,fff,fff□ffff,□
 GetTickCountffffŠ□,E,□,,,"□,,',,,,50",^□,,,^□
 dwSequence


```

IX
fff,xŽ,Š,,□•□ffff,“Žf□f,,,□□□fff,xŽ•Ā,•□ffff,□'Žf□f,,,□□□fff,xŽ•Ā,□'^□
IY
fff,yŽ,Š,,□•□ffff,“Žf□f,,,□□□fff,yŽ•Ā,•□ffff,□'Žf□f,,,□□□fff,y•Ā,□'^□
IZ
fff,zŽ,Š,,□•□ffff,“Žf□f,,,□□□fff,zŽ•Ā,•□ffff,□'Žf□f,,,□□□fff,zŽ•Ā,□'^□
fff,zŽ,,,,□,,'0,,,□
rgbButtons[4]
fff,□,'—□“,,fff,%o,,,,□□□□^fff,fff,,,□

```

```
DIOBJECTDATAFORMAT XE "DIOBJECTDATAFORMAT" §
```

```
typedef struct {
    const GUID * pguid;
```

```
    DWORD dwOfs;
```

```
    DWORD dwType;
```

```
    DWORD dwFlags;
```

```

}
DIOBJECTDATAFORMAT□fff,•□,DIOBJECTDATAFORMAT□“,, ,
Šfffff(Ž□fff,,),‘,,,,, “,“,,,,, ,,,, “,,”—
, IDirectInputDevice::SetDataFormat,“,,,DIDATAFORMAT□“Ž,□’ ,fff
□fff,DIOBJECTDATAFORMAT□“,”—, ,•—,,, ,, “,‘,
    c_dfDIMouse    c_dfDIKeyboard 0

```

pguid

```

Ž f f ff fff,—<, ,,,ff ff,NULL,fff,,, ,,,,fff,ffffff,,,,,,^—
,,,
dwOfs
“—f f,‘,
f ffff“,,,ffff, Ž,,,,,DWORDfff,f f,,,,,4,” ,,,,,, ,,,, fff,‘,,*,,,,
fff fffffff,’

```

dwType

```

fffff,‘ ,,,fff□fffŽ’Ž□,,□fffffffff(Ž□fff,,)
,‘ ,,,DIDFT_*fff,‘, ,,, ^fff,,ff DIDFT_ANYINSTANCE

```

dwFlags

0 0

0

DirectInput

4 DWORD

DIOBJECTDATAFORMAT dfAnyAxis = {

```
    0, // fffff□f
```

```
    4, // 0,,,,,,
```

```
};
```

```

Ž,ffffff□f□f□ff□fffŽ',,□ffff,XŽ,ffff□f□f,,,ffffff12,DWORD,Š",,□ffff,•□,XŽ,□,,
□□□□□,—%””,“,“,□
DIOBJECTDATAFORMAT dfAnyXAxis = {
    &GUID_XAxis, // XŽ,,,,,,
    12, //
    DIDFT_AXIS | DIDFT_ANYINSTANCE, // の X
    0, // 0
};

```

の DirectInput
 の の BYTE の 16

```

DIOBJECTDATAFORMAT dfAnyButton = {
    0, //
    16, // ff
    DIDFT_BUTTON | DIDFT_ANYINSTANCE, // ,,fff,,,,,,
    0, // 0
};

```

```

Ž,ffffff f f ff fffŽ',, DirectInput, ,—
%”,ÒFire”fff,“, ,’,BYTE, ^f
fff17,’,,
fff,ÒFire”fff,‘ ,,, ,f f ff fff,fff•%”,,,

```

```

DIOBJECTDATAFORMAT dfAnyButton = {
    &GUID_FireButton, // ffffffff
    17, //
    DIDFT_BUTTON |
    DIDFT_ANYINSTANCE, // の
    0, // 0
};

```

の , の 0
 18 BYTE の

```

0
DIOBJECTDATAFORMAT dfButton0 = {
    0, //
    18, //
    DIDFT_BUTTON |
    DIDFT_MAKEINSTANCE(0), // 0
    0, // 0
};

```

DIPROPDWORD XE "DIPROPDWORD" §


```

□•Ž•Ž□
szPname
ffffffff,□•-,Š,,,□NULL,□,,•Ž—□
wXmin□wXmax
x□',□□',,,,□"□
wYmin□wYmax
y□',□□',,,,□"□
wZmin□wZmax
z□',□□',,,,□"□
wNumButtons
ffffffff.fff□□
wPeriodMin□wPeriodMax
ffff□fff,ffffff

```

wRmin wRmax
 の 4 の

wUmin wUmax

u (5 の) の

wVmin wVmax

v (6 の) の

wCaps
 のの

JOYCAPS_HASPOV

JOYCAPS_HASR

(4 の)

JOYCAPS_HASU

u (5 0)

JOYCAPS_HASV

ffffffff,v (6"-Ž) •,Ž,

JOYCAPS_HASZ

ffffffff,z • •,Ž,

JOYCAPS_POV4DIR

,ff f,, '‰ ' Ć ,,,‰

JOYCAPS_POVCTS

ffffffff,Ž“~“•Š,ff f,,

wMaxAxes

ffffffff,ff f,,Ž, ‘

wNumAxes

ffffffff,,Ć Ž—,,,,Ž

wMaxButtons

ffffffff,ff f,,fff, ‘

szRegKey

ffffffff,‘,fffff f ,Š, NULL, ,,Ž—

szOEMVxD

ffffffffffffOEM,Ž•,, NULL, ,,Ž—

Ž joyGetDevCaps

JOYINFOEX XE "JOYINFOEX" §

typedef struct joyinfoex_tag {

DWORD dwSize;

DWORD dwFlags;

```

    DWORD dwXpos;
    DWORD dwYpos;
    DWORD dwZpos;
    DWORD dwRpos;
    DWORD dwUpos;
    DWORD dwVpos;
    DWORD dwButtons;
    DWORD dwButtonNumber;
    DWORD dwPOV;
    DWORD dwReserved1;
    DWORD dwReserved2;
} JOYINFOEX;

```

の の

```

dwSize
の の
dwFlags
の
0 i の
JOY_RETURNALL

```

i

```

JOY_RETURNRAWDATA
i

```

```

JOY_RETURN

```

```

JOY_RETURNBUTTONS

```

```

dwButtons

```

```

JOY_RETURNCENTERED

```

```

ののの

```

```

JOY_RETURNPOV

```

```

dwPOV

```

```

JOY_RETURNPOVCTS

```

```

The dwPOV 100 1

```

```

JOY_RETURNR

```


□□□□□□□□□□,,□'9,000,90.00“,•CE,•,□

Windows 95,ffffff,fffffffffffff,□CEŽ“,,,,i',Š“,ffff□fff,—,,,,,□□□ —
 %o”,,,,~“„f□f,JOY_RETURNPOVCTSfff,—
 „Ž“,,,,,i,,JOY_RETURNPOVCTSfff,□JOY_RETURNPOVfff,—,,,,JOY
 POV”□,ff□f,,,□
 Ž□□joyGetPosEx

ff□ff•□
 c_dfDIKeyboard XE "c_dfDIKeyboard" §
 ,,,,,'<,,,,,DIDATAFORMAT の
 DINPUT.LIB

の IDirectInputDevice::SetDataFormat

c_dfDIMouse XE "c_dfDIMouse" §

DIDATAFORMAT
 の DINPUT.LIB

の IDirectInputDevice::SetDataFormat

GUID_SysKeyboard XE "GUID_SysKeyboard" §

DirectInput の GUID

,', fffff f f,',,fff ffff, ,,,,,IDirectInput::CreateDevice,“,,,,,,
 ,

GUID_SysMouse XE "GUID_SysMouse" §

,,,,,'<,,,,,DirectInput,ffffffGUID, fffff,ffffff,,Ž , ,', fffff
 f,',,fff ffff, ,,,,,IDirectInput::CreateDevice,“,,,,,,

,

DIPROP_AXISMODE XE "DIPROP_AXISMODE" §

,,,,,'<,,,,,fffff, Žf ff f,fff, の
 ,ffffff,',,,,,fff“,—,,,,, dwHowff ff,DIPH_DEVICE,,,,,,
 ,,fffff,DIPROPDWORD “,—,
DIPROPDWORD “,pdiphff ff,DIPROPHEADER “,ffff,,,,,, d
 wDataaff ff,Žf f,Š,, ,,,Ž,Ž,,,,,
 DIPROPHEADER “,dwObjff ff,0,,,,,, ,,,,,fffff ’,“,ffffff,,,,ff
 ff“,—,,,,,Ž, dwSize
 DIPROPDWORD “,fff,fff,,,,,,
 DIPROPDWORD “,dwDataaff ff,^%o,’,,,,
DIPROPAXISMODE_ABS: ‘ •,Ž~,’,,, Ž~“,ŽŠ,,,,—E,,,

DIK_9 9 f□
 DIK_0 0 f□
 DIK_MINUS *ffff*□f□f, - f□
 DIK_EQUALS = f□
 DIK_BACK Backspacef□
 DIK_TAB Tabf□
 DIK_Q Q f□
 DIK_W W f□
 DIK_E E f□
 DIK_R R f□
 DIK_T T f□
 DIK_Y Y f□
 DIK_U U f□
 DIK_I I
 DIK_O O
 DIK_P P
 DIK_LBRACKET [
 DIK_RBRACKET]
 DIK_RETURN Enter
 DIK_LCONTROL Ctrl
 DIK_A A
 DIK_S S
 DIK_D D
 DIK_F F
 DIK_G G
 DIK_H H
 DIK_J J f
 DIK_K K f
 DIK_L L f
 DIK_SEMICOLON ; f
 DIK_APOSTROPHE *ffffff*□ ()
 DIK_GRAVE accent grave (`) f
 DIK_LSHIFT Shiftf
 DIK_BACKSLASH \ f
 DIK_Z Z f
 DIK_X X f
 DIK_C C f
 DIK_V V f
 DIK_B B f□
 DIK_N N
 DIK_M M f

DIK_COMMA , f
 DIK_PERIOD ffff f f, .f

DIK_SLASH

DIK_RSHIFT
 DIK_MULTIPLY
 DIK_LMENU
 DIK_SPACE
 DIK_CAPITAL
 DIK_F1
 DIK_F2

ffff□*f*□*f*, / *f*□

Shift

の *

Alt

Caps Lock

F1

F2

DIK_F3 F3 f□

DIK_F4 F4 f□

DIK_F5 F5 f□

DIK_F6 F6 f□

DIK_F7 F7 f□

DIK_F7 *fff*□,7*f*□

DIK_NUMPAD8 *fff*□,8*f*□

DIK_NUMPAD9 *fff*□,9*f*□

DIK_SUBTRACT *fff*□, - *f*□

DIK_NUMPAD4 *fff*□,4*f*□

DIK_NUMPAD5 *fff*□,5*f*□

DIK_NUMPAD6 *fff*□,6*f*□

DIK_ADD *fff*□,+ *f*□

DIK_NUMPAD1 *fff*□,1*f*□

DIK_NUMPAD2 の 2

DIK_NUMPAD3 の 3

DIK_NUMPAD0 の 0

DIK_DECIMAL の .

DIK_F11 F11

DIK_F12 F12

DIK_NUMPADENTER の Enter

DIK_RCONTROL Ctrl

DIK_DIVIDE の /

DIK_SYSRQ SysRq

DIK_RMENU Alt

DIK_HOME Home

DIK_UP
DIK_PRIOR PgUp
DIK_LEFT
DIK_RIGHT
DIK_END End
DIK_DOWN
DIK_NEXT PgDn

DIK_INSERT Insertf
DIK_DELETE Deletf
DIK_LWIN Windowsf
DIK_RWIN %oWindowsf
DIK_APPS AppMenuf

fff ffff' XE "fff ffff" §

DIMOFS_BUTTON0

DIMOUSESTATE “, “,’,,ffffff0, ‘,ffff ,’, DIDEVICEOBJECT
DATA “,dwOfsff ff,,*, f f,ffffff0,“—,,,,,Ž,

DIMOFS_BUTTON1

DIMOUSESTATE “, “,’,,ff 1 のの
,’, DIDEVICEOBJECTDATA “,dwOfsff ff,,*, f f,ffffff1,“—
,,,,,Ž,

DIMOFS_BUTTON2

DIMOUSESTATE□“, “,’,,fff 2 ののDIDEVICEOBJECTDATA
の dwOfs 2

DIMOFS_BUTTON3

DIMOUSESTATE の 3 ののDIDEVICEOBJECTDATA の dwOfs

3

DIMOFS_X

DIMOUSESTATE □ □ x ののの

DIDEVICEOBJECTDATA □ dwOfs の x の

,,,,,,Ž,□

DIMOFS_Y

DIMOUSESTATE□“,□’,,fff,y•CE,^,ffffff□,,’□DID□’,,fff,z•CE,^,ffffff□,,’□DIDEVICE
OBJECTDATA□“,dwOfsff□ff,,,□f□f,fff,z•CE,^,“—,,,,,Ž,□

DirectInputffff□fff<□f□f XE "DirectInputffff□fff<□f□f" §
DirectInputffff□fff< f□f,DIDEVICEINSTANCE

DIDEVTYPE_MOUSE: ()

“Žffff,,□‘ •; ,,, ,,, ’,, ‘ •; ffff,”Ž,,,,Š,,,ffff,,’,,“ •, Ć,,
,□Ć%0,,□‘Žffffff,,“,,‘•

DIDFT_ABSAXIS

の

の

の 20 の の 20 の
DIPROP_GRANULARITY

DIDFT_AXIS

の IDirectInputDevice::EnumObjects のの

DIDFT_PSHBUTTON

DIDFT_TGLBUTTON

DIDFT_BUTTON

ののの

DirectInputŠ ,,,•,,,ff ,—◁,,, ŠŠ ,•,,,ff f f,fff,,,, Ć ,Š ,’Ž,Ž
,,,

JOYERR_NOCANDO

ffffff,ffff,,ffff •, ,,,

JOYERR_NOERROR

—◁, —,,

JOYERR_REGISTRYNOTVALID

ffff,,^ ,ffffffffffff,—Ć,f f,Š,,

JOYERR_UNPLUGGED

Ž',,,fffffff,fff□f,□',,,,,,□

MMSYSERR_BADDEVICEID

Ž',,,fffffff,Ž•Ž,-CE,,,i

MMSYSERR_INVALIDPARAM

-CE,fff□f,“,,,□

MMSYSERR_NODRIVER

fffffff,‘□,,,□