This call reads an event from the mouse device FIFO event queue, and places it in a structure provided by the application.

Syntax

MouReadEventQue (Buffer, ReadType, DeviceHandle)

Parameters

;Buffer (PMOUEVENTINFO) - output: Address of the status of the mouse event queue. ;DeviceHandle (HMOU) - input : Handle of the mouse device from a previous MouOpen.

Return Code

;rc (USHORT) - return:Return code descriptions are: * 0 NO_ERROR * 385 ERROR_MOUSE_NO_DEVICE * 387 ERROR_MOUSE_INV_PARMS * 393 ERROR_MOUSE_NO_DATA * 466 ERROR_MOU_DETACHED * 501 ERROR_MOUSE_NO_CONSOLE * 505 ERROR_MOU_EXTENDED_SG

Remarks

The types of queued events are directly affected by the current value of the Mouse EventMask. MouSetEventMask is used to indicate the types of events desired, and MouGetEventMask is used to query the current value of the mask. Refer to these functions for further explanation of the masking of events.

Recognition of the mouse transition depends on the use of MouState returned in the event record. The application should focus on bit transitions that occur in this word. It is important to properly set the event mask with MouSetEventMask for reporting the state transitions.

MouState reports the state of the mouse that resulted from the action that caused the event. The action can be pressing or releasing a button, and/or moving the mouse. All status is given, regardless of the EventMask that was used to determine whether or not to report the event.

For example, assume the EventMask indicates that the application wishes only button 1 event. The EventMask has only bits 1 and 2 set in this case. Also assume the current state of the mouse is no buttons down, and mouse is not moving. At this point, button 1 is pressed causing an event; the status shows button 1 down (bit 2 set). Next the mouse is moved, thereby causing more events; status shows bit 1 set. Finally, mouse is stopped and button 1 is released. The event shows status with no bits set.

Next, button 2 is pressed. No event occurs. Mouse is then moved; again, no event. Then, while mouse is still in motion, button 1 is pressed; an event is generated with bits 1 and 3 set in the state word. While mouse is still in motion, both buttons are released. Because button 1 changes states, an event occurs. The state word has bit 0 set. Finally, mouse is stopped. No event occurs, again because no button 1 transition has taken place.

The Row and Column fields in the Buffer Parameter may contain either absolute display coordinates or relative mouse motion in mickeys. See MouSetDevStatus for additional information.

Bindings

C

<PRE> typedef struct _MOUEVENTINFO { /* mouev */

```
USHORT fs;                     /* State of mouse at time event was reported */
ULONG  time;                   /* Time since boot in milliseconds */
USHORT row;                    /* Absolute/relative row position */
USHORT col;                    /* Absolute/relative column position */
```

}MOUEVENTINFO;

#define INCL_MOU

USHORT rc = MouReadEventQue(Buffer, ReadType, DeviceHandle);

PMOUEVENTINFO Buffer; /* 10 byte Structure address */ PUSHORT ReadType; /* Read type */ HMOU DeviceHandle; /* Mouse device handle */

USHORT rc; /* return code */ </PRE>

MASM

<PRE> MOUEVENTINFO struc

```
mouev_fs   dw  ?  ;State of mouse at time event was reported
mouev_time dd  ?  ;time since boot in milliseconds
mouev_row  dw  ?  ;absolute/relative row position
mouev_col  dw  ?  ;absolute/relative column position
```

MOUEVENTINFO ends

EXTRN MouReadEventQue:FAR INCL_MOU EQU 1

PUSH@ OTHER Buffer ;10 byte Structure address PUSH@ WORD ReadType ;Read type PUSH WORD DeviceHandle ;Mouse device handle CALL MouReadEventQue

Returns WORD </PRE>

Mou