MouReadEventQue

Bindings: C, MASM

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

MouReadEventQue (Buffer, ReadType, DeviceHandle)

Buffer (**PMOUEVENTINFO**) - output Address of the status of the mouse event queue.

moustate (**USHORT**) State of the mouse at the time of the event.

Bit	Description
15-7	Reserved, set to zero.
6	Set if button 3 is down.
5	Set if mouse is moving and button 3 is down.
4	Set if button 2 is down.
3	Set if mouse is moving and button 2 is down.
2	Set if button 1 is down.
1	Set if mouse is moving and button 1 is down.
0	Set if mouse is moving and no buttons are down.

eventtime (**ULONG**) Time stamp (in milliseconds) since the system was started.

row (**USHORT**) Absolute or relative row position.

col (**USHORT**) Absolute or relative column position.

ReadType (**PUSHORT**) - input Address of the action to take when MouReadEventQue is issued and the mouse event queue is empty. If the mouse event queue is not empty, this parameter is not examined by the mouse support. *ReadType* values are:

Value	Definition
0	No Wait for data on empty queue (return a NULL record)
1	WAIT for data on empty queue.

DeviceHandle (**HMOU**) - input Handle of the mouse device from a previous MouOpen.

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.