MouRegister

Bindings: C, MASM

This call registers a mouse subsystem within a session.

MouRegister (ModuleName, EntryName, Mask)

ModuleName (**PSZ**) - input Address of the dynamic link module name. The maximum length is 9 bytes (including ASCIIZ terminator).

*EntryName?? (**PSZ**) - input Address of the dynamic link entry point name of a routine that receives control when any of the registered functions are called. The maximum length is 33 bytes (including ASCIIZ terminator).* Mask *(**ULONG**) - input A mask of bits, where each bit set to 1 identifies a mouse function being registered. Bit values are: ^Bit ^Description ^ |31-22 |Reserved, set to zero | |21 |MouSetDevStatus | |20 |MouFlushQue | |19 |MouInitReal | |18 |MouSetPtrPos | |17 |MouGetPtrPos | |16 |MouRemovePtr | |15 |MouDrawPtr | |14 |MouSetPtrShape | |13 |MouGetPtrShape | |12 |MouClose | |11 |MouOpen | |10 |Reserved | |9 |Reserved | |8 |MouSetEventMask | |7 |MouSetScaleFact | |6 |MouGetEventMask | |5 |MouGetScaleFact | |4 |MouReadEventQue | |3 |MouGetNumQueEl | |2 |MouGetDevStatus | |1 |MouGetNumMickeys | |0 |MouGetNumButtons |* rc *(**USHORT**) - return Return code descriptions are: |0 |NO_ERROR | |385 |ERROR_MOUSE_NO_DEVICE | |413 |ERROR_MOUSE_INVALID_ASCIIZ | |414 |ERROR_MOUSE_INVALID_MASK | |415 |ERROR_MOUSE_REGISTER | |466 |ERROR_MOU_DETACHED | |505 |ERROR_MOU_EXTENDED_SG | **Remarks** The Base Mouse Subsystem is the default mouse subsystem. There can be only one MouRegister outstanding for each session without an intervening MouDeRegister. MouDeRegister must be issued by the same process that issued MouRegister. When any registered function is called, control is routed to EntryName. When this routine is entered, four additional values are pushed onto the stack. The first is the index number (Word) of the function being called. The second is a near pointer (Word). The third is the caller's DS register (Word). The fourth is the return address (DWord) to the mouse router. For example, if MouGetNumMickeys were called and control routed to EntryName, the stack would appear as if the following instructions were executed: <code asm> PUSH@ WORD NumberOfMickeys PUSH WORD DeviceHandle CALL FAR MouGetNumMickeys PUSH WORD Function Code CALL NEAR Entry point in Mouse Router PUSH DS CALL FAR EntryName. </code> When a registered function returns to the Mouse Router, AX is interpreted as follows: AX = 0 No error. Do not invoke the Base Mouse Subsystem routine. Return AX = 0. AX = -1 Invoke the BaseMouse Subsystem routine. Return AX = return code from the Base Mouse Subsystem. AX = error (if not 0 or -1) Do not invoke the Base Mouse Subsystem Routine. Return AX = error. When the mouse router receives a mouse call, it routes it to the Base Mouse Subsystem unless an application or other mouse subsystem has previously issued MouRegister for that call. If the call was registered, the subsystem is entered at the EntryName specified, and provided with the applicable function code. The registered function mask is used to determine whether a requested function is performed by the registered mouse subsystem or default to the Base Mouse Subsystem. The following list shows the relationship of the mouse API calls and the Function Code passed to either the Base Mouse Subsystem or a registered mouse subsystem. ^MOU API calls ^Function Code ^ |MouGetNumButtons |00H | |MouGetNumMickeys |01H | |MouGetDevStatus |02H | |MouGetNumQueEl |03H | |MouReadEventQue |03H | |MouGetScaleFact |05H | |MouGetEventMask |06H | |MouSetScaleFact |07H | |MouSetEventMask |08H | |Reserved |09H | |Reserved |0AH | |MouOpen |0BH | |MouClose |0CH | |MouGetPtrShape |0DH | |MouSetPtrShape |0EH | |MouDrawPtr |0FH | |MouRemovePtr |10H | |MouGetPtrPos |11H | |MouSetPtrPos |12H | |MouInitReal |13H | |MouFlushQue |14H | |MouSetDevStatus |15H | A registered mouse sybsystem must leave the stack, on exit, in the exact state it was received.*