

This call registers a mouse subsystem within a session.

## Syntax

MouRegister (ModuleName, EntryName, Mask)

## Parameters

;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.

## Return Code

;rc (USHORT) - return:Return code descriptions are: \* 0 NO\_ERROR \*385 ERROR\_MOUSE\_NO\_DEVICE \*413 ERROR\_MOUSE\_INVALID\_ASCII \*414 ERROR\_MOUSE\_INVALID\_MASK \*415 ERROR\_MOUSE\_REGISTER \*466 ERROR\_MOUSE\_DETACHED \*505 ERROR\_MOUSE\_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 MouGetNumMickeyes were called and control routed to EntryName, the stack would appear as if the following instructions were executed: PUSH@ WORD NumberOfMickeyes PUSH WORD DeviceHandle CALL FAR MouGetNumMickeyes PUSH WORD Function Code CALL NEAR Entry point in Mouse Router PUSH DS CALL FAR EntryName. 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. {

| Function Code | Mouse API Call              |
|---------------|-----------------------------|
| 0             | MOU_API_NO_ERROR            |
| 385           | MOU_API_MOUSE_NO_DEVICE     |
| 413           | MOU_API_MOUSE_INVALID_ASCII |
| 414           | MOU_API_MOUSE_INVALID_MASK  |
| 415           | MOU_API_MOUSE_REGISTER      |
| 466           | MOU_API_MOUSE_DETACHED      |
| 505           | MOU_API_MOUSE_EXTENDED_SG   |

 !MOU API

calls||Function Code||Function Mask

|                  |     |
|------------------|-----|
| MouGetNumButtons | 00H |
| MouGetNumMickeys | 01H |
| MouGetDevStatus  | 02H |
| MouGetNumQueEl   | 03H |
| MouReadEventQue  | 04H |
| MouGetScaleFact  | 05H |
| MouGetEventMask  | 06H |
| MouSetScaleFact  | 07H |
| MouSetEventMask  | 08H |
| MouGetHotKey     | 09H |
| MouSetHotKey     | 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.

## Bindings

## C

```
<PRE> #define INCL_MOU
```

```
USHORT rc = MouRegister(ModuleName, EntryName, Mask);
```

```
PSZ ModuleName; /* Module Name */ PSZ EntryName; /* Entry Name */ ULONG Mask; /* Function Mask */
```

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

## MASM

<PRE> EXTRN MouRegister:FAR INCL\_MOU EQU 1

PUSH@ ASCIIZ ModuleName ;Module Name PUSH@ ASCIIZ EntryName ;Entry Name PUSH DWORD  
Mask ;Function Mask CALL MouRegister

Returns WORD </PRE>

## Related Functions

\* [MouDeRegister](#)

[Mou](#)

From:

<http://ftp.osfree.org/doku/> - **osFree wiki**

Permanent link:

<http://ftp.osfree.org/doku/doku.php?id=en:docs:fapi:mouregister&rev=1633526347>

Last update: **2021/10/06 13:19**

