 This is part of **Family API** which allow to create dual-os version of program runs under OS/2 and DOS

Note: This is legacy API call. It is recommended to use 32-bit equivalent

2021/09/17 04:47 · prokushev · 0 Comments
2021/08/20 03:18 · prokushev · 0 Comments

MouOpen

This call opens the mouse device for the current session.

Syntax

```
MouOpen (DriverName, DeviceHandle)
```

Parameters

- DriverName (PSZ) - input : DriverName is a far pointer to an ASCIIZ string in application storage containing the name of the pointer draw device driver to be used as the pointer-image drawing routine for this session.
  - The name of the device driver must be included in the CONFIG.SYS file at system start-up time. Applications that use the default pointer draw device driver supplied by the system must push a double-word of 0s in place of an address.
  - DriverName has a different definition when the caller is the Base Video Subsystem (BVS). In this case the selector portion of the far address is zero. The offset portion is non-zero and contains a display configuration number (sequentially numbered where 1 is the first display configuration). The MouOpen call issued by BVS is executed on the VioSetMode path. Using the display configuration number passed on the MouOpen call, the Base Mouse Subsystem can detect a change in display configurations. This form of the MouOpen call is not recommended for applications. Applications should either push the far address of an ASCIIZ pointer draw device driver name or push two words of zeros.
- DeviceHandle (PHMOU) - output :Address of a 1-word value that represents the mouse handle returned to the application.

Return Code

rc (USHORT) - return

Return code descriptions are:

- 0 NO_ERROR
- 385 ERROR_MOUSE_NO_DEVICE

- 390 ERROR_MOUSE_INV_MODULE_PT
- 466 ERROR_MOU_DETACHED
- 501 ERROR_MOUSE_NO_CONSOLE
- 505 ERROR_MOU_EXTENDED_SG

Remarks

MouOpen initializes the Mouse functions to a known state. The application may have to issue additional mouse functions to establish the environment it desires. For example, after the MouOpen, the collision area is defined to be the size of the entire display. Therefore, to get the pointer to be displayed, the application must issue a MouDrawPtr to remove the collision area.

The state of the mouse after the first MouOpen is:

- Row/Col scale factors set to 16/8. (See MouSetScaleFact)
- All events reported. (See MouSetEventMask)
- Empty event queue. (See MouReadEventQue and MouGetNumQueEl)
- All user settable Device Status bits reset. (Set to zero. See MouSetDevStatus)
- Pointer set to center of screen if valid display mode is set. (See MouSetPtrPos)
- Pointer shape set to the default for the pointer device driver currently registered in the session. (See MouSetPtrShape)
- Collision area equal to full screen. (See MouDrawPtr and MouRemovePtr)

Bindings

C

```
#define INCL_MOU

USHORT  rc = MouOpen(DriverName, DeviceHandle);

PSZ     DriverName;     /* Pointer draw driver name */
PHMOU   DeviceHandle;   /* Mouse device handle */

USHORT  rc;             /* return code */
```

MASM

```
EXTRN     MouOpen:FAR
INCL_MOU  EQU 1

PUSH@  ASCIIZ  DriverName    ;Pointer draw driver name
PUSH@  WORD    DeviceHandle  ;Mouse device handle
CALL   MouOpen
```

```
Returns WORD
```

Related Functions

MouClose