

input

COLLABORATORS

	<i>TITLE :</i> input		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	input	1
1.1	input.doc	1
1.2	input.device/AddHandler	1
1.3	input.device/RemHandler	1
1.4	input.device/Reset	2
1.5	input.device/SetMPort	2
1.6	input.device/SetMTrig	3
1.7	input.device/SetMType	3
1.8	input.device/SetPeriod	4
1.9	input.device/SetThresh	4
1.10	input.device/Start	4
1.11	input.device/WriteEvent	5

Chapter 1

input

1.1 input.doc

AddHandler()	SetMPort()	SetPeriod()	WriteEvent()
RemHandler()	SetMTrig()	SetThresh()	
Reset()	SetMType()	Start()	

1.2 input.device/AddHandler

NAME

AddHandler - add an input handler to the device

FUNCTION

Add a function to the list of functions called to handle input events generated by this device. The function is called as

```
newInputEvents = Handler(inputEvents, handlerData);
D0                                A0                                A1
```

IO REQUEST

io_Message	mn_ReplyPort set
io_Device	preset by OpenDevice
io_Unit	preset by OpenDevice
io_Command	IND_ADDHANDLER
io_Data	a pointer to an interrupt structure.
is_Data	the handlerData pointer described above
is_Code	the Handler function address

NOTES

The interrupt structure is kept by the input device until a RemHandler command is satisfied for it.

1.3 input.device/RemHandler

NAME

RemHandler - remove an input handler from the device

FUNCTION

Remove a function previously added to the list of handler functions.

IO REQUEST

io_Message	mn_ReplyPort set
io_Device	preset by OpenDevice
io_Unit	preset by OpenDevice
io_Command	IND_REMHANDLER
io_Data	a pointer to the interrupt structure.

NOTES

This command is not immediate

1.4 input.device/Reset

NAME

Reset - reset the input device

FUNCTION

Reset resets the input device without destroying handles to the open device.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Unit	preset by the call to OpenDevice
io_Command	CMD_RESET
io_Flags	IOB_QUICK set if quick I/O is possible

1.5 input.device/SetMPort

NAME

SetMPort - set the current mouse port

FUNCTION

This command sets the gameport port at which the mouse is connected.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Unit	preset by the call to OpenDevice
io_Command	IND_SETMPORT
io_Flags	IOB_QUICK set if quick I/O is possible
io_Length	1
io_Data	a pointer to a byte that is either 0 or 1, indicating that mouse input should be obtained

from either the left or right controller port,
respectively.

1.6 input.device/SetMTrig

NAME

SetMTrig - set the conditions for a mouse port report

FUNCTION

This command sets what conditions must be met by a mouse before a pending Read request will be satisfied. The trigger specification is that used by the gameport device.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Unit	preset by the call to OpenDevice
io_Command	IND_SETMTRIG
io_Flags	IOB_QUICK set if quick I/O is possible
io_Length	sizeof(gameportTrigger)
io_Data	a structure of type GameportTrigger, which has the following elements
gpt_Keys -	
	GPTB_DOWNKEYS set if button down transitions trigger a report, and GPTB_UPKEYS set if button up transitions trigger a report
gpt_Timeout -	
	a time which, if exceeded, triggers a report; measured in vertical blank units (60/sec)
gpt_XDelta -	
	a distance in x which, if exceeded, triggers a report
gpt_YDelta -	
	a distance in x which, if exceeded, triggers a report

1.7 input.device/SetMType

NAME

SetMType - set the current mouse port controller type

FUNCTION

This command sets the type of device at the mouse port, so the signals at the port may be properly interpreted.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Unit	preset by the call to OpenDevice
io_Command	IND_SETMTYPE
io_Flags	IOB_QUICK set if quick I/O is possible
io_Length	1

io_Data the address of the byte variable describing
the controller type, as per the equates in
the gameport include file

1.8 input.device/SetPeriod

NAME

SetPeriod - set the key repeat period

FUNCTION

This command sets the period at which a repeating key repeats.

This command always executes immediately.

IO REQUEST - a timerequest

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Unit	preset by the call to OpenDevice
io_Command	IND_SETPERIOD
io_Flags	IOB_QUICK set if quick I/O is possible
io_tv_Secs	the repeat period seconds
io_tv_Micro	the repeat period microseconds

1.9 input.device/SetThresh

NAME

SetThresh - set the key repeat threshold

FUNCTION

This command sets the time that a key must be held down before it can repeat. The repeatability of a key may be restricted (as, for example, are the shift keys).

This command always executes immediately.

IO REQUEST - a timerequest

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Unit	preset by the call to OpenDevice
io_Command	IND_SETTHRESH
io_Flags	IOB_QUICK set if quick I/O is possible
io_tv_Secs	the threshold seconds
io_tv_Micro	the threshold microseconds

1.10 input.device/Start

NAME

Start - restart after stop

FUNCTION

Start restarts the unit after a stop command.

```
IO REQUEST
  io_Message      mn_ReplyPort set if quick I/O is not possible
  io_Device       preset by the call to OpenDevice
  io_Unit         preset by the call to OpenDevice
  io_Command      CMD_START
  io_Flags        IOB_QUICK set if quick I/O is possible
```

1.11 input.device/WriteEvent

NAME

WriteEvent - propagate input event(s) to all handlers

FUNCTION

```
IO REQUEST
  io_Message      mn_ReplyPort set if quick I/O is not possible
  io_Device       preset by the call to OpenDevice
  io_Unit         preset by the call to OpenDevice
  io_Command      IND_WRITEEVENT
  io_Flags        IOB_QUICK set if quick I/O is possible
  io_Length       the size of the io_Data area in bytes: there
                  are sizeof(inputEvent) bytes per input event.
  io_Data         a buffer area with input events(s). The
                  fields of the input event are:
    ie_NextEvent
                  links the events together, the last event
                  has a zero ie_NextEvent.
    ie_Class
    ie_SubClass
    ie_Code
    ie_Qualifier
    ie_X, ie_Y
    ie_TimeStamp
                  as desired
```

NOTES

The contents of the input event(s) are destroyed.