input

COLLABORATORS

	TITLE : input	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 23, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

input

1.1 input.doc

```
    IND_ADDHANDLER
    IND_REMHANDLER
    IND_SETMPORT
    IND_SETMTRIG
    IND_SETMTYPE
    IND_SETPERIOD
    IND_SETTHRESH
    IND_WRITEEVENT
    PeekQualifier()
```

1.2 input.device/IND_ADDHANDLER

```
 NAME
IND_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/IND_REMHANDLER

```
 NAME
IND_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/IND_SETMPORT

```
 NAME
IND_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.
```

1.5 input.device/IND_SETMTRIG

```
 NAME
IND_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
```

1.6 input.device/IND_SETMTYPE

```
 NAME
IND_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
```

1.7 input.device/IND_SETPERIOD

```
 NAME
IND_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
```

```
tr_node.io_Message  mn_ReplyPort set if quick I/O is not possible
tr_node.io_Device preset by the call to OpenDevice
tr_node.io_Unit   preset by the call to OpenDevice
tr_node.io_Command  IND_SETPERIOD
tr_node.io_Flags  IOB_QUICK set if quick I/O is possible
tr_time.tv_secs   the repeat period seconds
tr_time.tv_micro  the repeat period microseconds
```

1.8 input.device/IND_SETTHRESH

```
 NAME
IND_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
tr_node.io_Message  mn_ReplyPort set if quick I/O is not possible
tr_node.io_Device preset by the call to OpenDevice
tr_node.io_Unit   preset by the call to OpenDevice
tr_node.io_Command  IND_SETTHRESH
tr_node.io_Flags  IOB_QUICK set if quick I/O is possible
tr_time.tv_secs   the threshold seconds
tr_time.tv_micro  the threshold microseconds
```

1.9 input.device/IND_WRITEEVENT

```
 NAME
IND_WRITEEVENT -- Propagate an input event 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 should be sizeof(struct InputEvent)
io_Data   a pointer to the struct InputEvent:
    ie_NextEvent
    will be ignored.
    ie_Class
    ie_SubClass
    ie_Code
    ie_Qualifier
    ie_X, ie_Y
    as desired
```

```
    ie_TimeStamp
    will be set by this call (V36)

 NOTES
The contents of the input event are destroyed.

This function was documented in V34 and earlier to allow
chaining of events via ie_NextEvent.  The implementation
never allowed that.  The documentation now reflects this.

ie_TimeStamp is set only in V36 and later.  Software written
to run on earlier versions should set this field to the
current time.
```

1.10 input.device/PeekQualifier

```
 NAME
PeekQualifier -- get the input device's current qualifiers (V36)

 SYNOPSIS
qualifier = PeekQualifier()
d0

UWORD PeekQualifier( VOID );

 FUNCTION
This function takes a snapshot of what the input device thinks
the current qualifiers are.

 RESULTS
qualifier - a word with the following bits set according to
    what the input device knows their state to be:
  IEQUALIFIER_LSHIFT, IEQUALIFIER_RSHIFT,
  IEQUALIFIER_CAPSLOCK, IEQUALIFIER_CONTROL,
  IEQUALIFIER_LALT, IEQUALIFIER_RALT,
  IEQUALIFIER_LCOMMAND, IEQUALIFIER_RCOMMAND,
  IEQUALIFIER_LEFTBUTTON, IEQUALIFIER_RBUTTON,
  IEQUALIFIER_MIDBUTTON

 NOTE
This function is new for V36.

 SEE ALSO
devices/inputevent.h
```