NAME

usbhidaction - perform actions according to USB HID controls

SYNOPSIS

usbhidaction [**-diev**] **-c** *config-file* **-f** *device* [**-p** *pidfile*] [**-t** *tablefile*] *arg ...*

DESCRIPTION

The **usbhidaction** utility can be used to execute commands when certain values appear on HID controls. The normal operation for this program is to read the configuration file and then become a daemon and execute commands as the HID items specify. If a read from the HID device fails, the program dies; this will make it die when the USB device is unplugged.

The options are as follows:

-d Toggle the daemon flag.

-e Instruct **usbhidaction** to die early. Useful when specified with multiple verbose options to see how files are parsed.

-i Ignore HID items in the configuration file that do not exist in the device.

-v Be verbose, and do not become a daemon.

-c *config-file*
Specify a path name for the configuration file.

-t *tablefile*
Specify a path name for the HID usage table file.

-f *device*
Specify a path name for the device to operate on. If *device* is numeric, it is taken to be the USB HID device number. If it is a relative path, it is taken to be the name of the device under */dev*. An absolute path is taken to be the literal device pathname.

-p *pidfile*
Specify an alternate file in which to store the process ID.

The configuration file will be re-read if the process gets a SIGHUP signal.

CONFIGURATION

The configuration file has a very simple format. Each line describes an action; if a line begins with a whitespace, it is considered a continuation of the previous line. Lines beginning with ‘#’ are considered as comments.

Each line has four parts: a name of a USB HID item, a value for that item, a debounce value, and an action. There must be whitespace between the parts.

The item names are similar to those used by usbhidctl(1).

The value is simply a numeric value. When the item reports this value, the action will be performed. If the value is ‘*’, it will match any value.

The debounce value is an integer not less than 0. The value of 0 indicates that no debouncing should occur. A value of 1 will only execute the action when the state changes. Values greater than one specify that an action should be performed only when the value changes by that amount.

The action is a normal command that is executed with system(3). Before it is executed some substitution will occur: ‘$n’ will be replaced by the *n*th argument on the command line, ‘$V’ will be replaced by the numeric value of the HID item, ‘$N’ will be replaced by the name of the control, and ‘$H’ will be replaced by the name of the HID device.

FILES

/usr/share/misc/usb_hid_usages The HID usage table.

/var/run/usbaction.pid The default location of the PID file.

EXAMPLES

The following configuration file can be used to control a pair of Philips USB speakers with the HID controls on the speakers.

```
# Configuration for various Philips USB speakers
Consumer:Volume_Increment                 1 0 mixer -f $1 vol.volume=+1%
Consumer:Volume_Decrement                 1 0 mixer -f $1 vol.volume=-1%
Consumer:Mute                             1 0 mixer -f $1 vol.mute=^
Consumer:Channel_Top.Microsoft:Base_Up    1 0 mixer -f $1 bass.volume=+1%
Consumer:Channel_Top.Microsoft:Base_Down 1 0 mixer -f $1 bass.volume=-1%
```

A sample invocation using this configuration would be

```
usbhidaction -f /dev/uhid1 -c conf /dev/mixer1
```

The following example controls the mixer volume using a Logitech Wingman. Notice the debounce of 1 for buttons and 5 for the slider.

```
Button:Button_1       1 1      mixer vol.volume=+10%
Button:Button_2       1 1      mixer vol.volume=-10%
Generic_Desktop:Z * 5          mixer vol.volume=‘echo $V | awk ’{printf("%.02f", $$1/255)}’‘
```

SEE ALSO

usbhidctl(1), usbhid(3), uhid(4), usb(4)

HISTORY

The **usbhidaction** command first appeared in NetBSD 1.6. The **usbhidaction** command appeared in FreeBSD 5.1.