NAME

libinput-record - record kernel events

SYNOPSIS

libinput record [options] [*/dev/input/event0* **[***/dev/input/event1* **...]]**

DESCRIPTION

The **libinput record** tool records kernel events from a device and prints them in a format that can later be replayed with the **libinput replay(1)** tool. This tool needs to run as root to read from the device.

The output of this tool is YAML, see **FILE FORMAT** for more details. By default it prints to stdout unless an output file is provided. For example, these are valid invocations:

libinput record /dev/input/event3 touchpad.yml

libinput record recording.yml

libinput record --all all-devices.yml

libinput record /dev/input/event3 /dev/input/event4 tp-and-keyboard.yml

The events recorded are independent of libinput itself, updating or removing libinput will not change the event stream.

OPTIONS

If one or more device nodes are given, this tool opens those device nodes. Otherwise, a list of devices is presented and the user can select the device to record. If unsure, run without any arguments.

--help Print help

--all Record all */dev/input/event** devices available on the system. This option should be used in exceptional cases only, the output file is almost always too noisy and replaying the recording may not be possible. This option requires **--output-file** and no device nodes may be provided on the commandline.

--autorestart=s
Terminate the current recording after *s* seconds of device inactivity. This option requires that a **--output-file** is specified. The output filename is used as prefix, suffixed with the date and time of the recording. The timeout must be greater than 0.

-o filename.yml
--output-file=filename.yml
Specifies the output file to use. If **--autorestart** is given, the filename is used as prefix only. Where --output-file is not given and the first **or** last argument is not an input device, the first **or** last argument will be the output file.

--grab Exclusively grab all opened devices. This will prevent events from being delivered to the host system.

--show-keycodes
Show keycodes as-is in the recording. By default, common keys are obfuscated and printed as **KEY_A** to avoid information leaks.

--with-libinput
Record libinput events alongside device events. **THIS FEATURE IS EXPERIMENTAL.** See section **RECORDING LIBINPUT EVENTS** for more details.

--with-hidraw
Record hidraw events alongside device events. **DO NOT TYPE SENSITIVE DATA.** See **RECORDING HID REPORTS** for more details.

RECORDING MULTIPLE DEVICES

Sometimes it is necessary to record the events from multiple devices simultaneously, e.g. when an interaction between a touchpad and a keyboard causes a bug. **libinput record** records multiple devices with an identical time offset, allowing for correct replay of the interaction.

If multiple devices are recorded, an output filename must be provided.

All devices to be recorded must be provided on the commandline, an example invocation is:

libinput record -o tap-bug /dev/input/event3 /dev/input/event7

Note that when recording multiple devices, only the first device is printed immediately, all other devices and their events are printed on exit.

RECORDING LIBINPUT EVENTS

When the **--with-libinput** commandline option is given, **libinput-record** initializes a libinput context for the devices being recorded. Events from these contexts are printed alongside the evdev events. **THIS**

FEATURE IS EXPERIMENTAL.

The primary purpose of this feature is debugging and event analysis, no caller may rely on any specific format of the events.

Note that while libinput and **libinput-record** see the same events from the device nodes, no guarantee can be given about the correct order of events. libinput events may come in earlier or later than the events from the device nodes and for some devices, libinput may internally alter the event stream before processing.

Note that the libinput context created by **libinput-record** does not affect the running desktop session and does not (can not!) copy any configuration options from that session.

RECORDING HID REPORTS

When the **--with-hidraw** commandline option is given, **libinput-record** searches for the hidraw node(s) of the given devices and prints any incoming HID reports from those devices.

HID reports are **not obfuscated** and a sufficiently motivated person could recover the key strokes from the logs. Do not type passwords while recording HID reports.

FILE FORMAT

The output file format is in YAML and intended to be both human-readable and machine-parseable. Below is a short example YAML file, all keys are detailed further below.

Any parsers must ignore keys not specified in the file format description. The version number field is only used for backwards-incompatible changes.

```
version: 1
ndevices: 2
libinput:
  version: 1.10.0
system:
  os: "fedora:26"
  kernel: "4.13.9-200.fc26.x86_64"
  dmi: "dmi:bvnLENOVO:bvrGJET72WW(2.22):bd02/21/2014:svnLENOVO:..."
devices:
  - node: /dev/input/event9
```

```
evdev:
  # Name: Synaptics TM2668-002
  # ID: bus 0x1d vendor 0x6cb product 00 version 00
  # Size in mm: 97x68
  # Supported Events:
  # Event type 0 (EV_SYN)

  #.. abbreviated for man page ...

  #
  name: Synaptics TM2668-002
  id: [29, 1739, 0, 0]
  codes:
    0: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] # EV_SYN
    1: [272, 325, 328, 330, 333, 334, 335] # EV_KEY
    3: [0, 1, 24, 47, 48, 49, 52, 53, 54, 55, 57, 58] # EV_ABS
  absinfo:
    0: [0, 4089, 0, 0, 42]
    1: [0, 2811, 0, 0, 41]
    24: [0, 255, 0, 0, 0]
    47: [0, 4, 0, 0, 0]
    48: [0, 15, 0, 0, 0]
    49: [0, 15, 0, 0, 0]
    52: [0, 1, 0, 0, 0]
    53: [0, 4089, 0, 0, 42]
    54: [0, 2811, 0, 0, 41]
    55: [0, 2, 0, 0, 0]
    57: [0, 65535, 0, 0, 0]
    58: [0, 255, 0, 0, 0]
  properties: [0, 2, 4]
hid: [12, 23, 34, 45, ...]
udev:
  properties:
  - ID_INPUT_MOUSE=1
  - ID_INPUT=1
quirks:
  - ModelAppleTouchpad=1
  - AttrSizeHint=32x32
events:
  - hid:
    time: [ 0, 0]
       hidraw0: [1, 2, 3, 4]
  - evdev:
    - [  0,      0,   3,  57,  1420] # EV_ABS / ABS_MT_TRACKING_ID   1420
    - [  0,      0,   3,  53,  1218] # EV_ABS / ABS_MT_POSITION_X    1218
    - [  0,      0,   3,  54,  1922] # EV_ABS / ABS_MT_POSITION_Y    1922
    - [  0,      0,   3,  52,     0] # EV_ABS / ABS_MT_ORIENTATION      0
    - [  0,      0,   3,  58,    47] # EV_ABS / ABS_MT_PRESSURE        47
    - [  0,      0,   1, 330,     1] # EV_KEY / BTN_TOUCH               1
    - [  0,      0,   1, 325,     1] # EV_KEY / BTN_TOOL_FINGER         1
    - [  0,      0,   3,   0,  1218] # EV_ABS / ABS_X                1218
    - [  0,      0,   3,   1,  1922] # EV_ABS / ABS_Y                1922
    - [  0,      0,   3,  24,    47] # EV_ABS / ABS_PRESSURE           47
    - [  0,      0,   0,   0,     0] # ------------ SYN_REPORT (0) ------- +0ms
  - evdev:
    - [  0,  11879,   3,  53,  1330] # EV_ABS / ABS_MT_POSITION_X    1330
    - [  0,  11879,   3,  54,  1928] # EV_ABS / ABS_MT_POSITION_Y    1928
    - [  0,  11879,   3,  58,    46] # EV_ABS / ABS_MT_PRESSURE        46
    - [  0,  11879,   3,   0,  1330] # EV_ABS / ABS_X                1330
    - [  0,  11879,   3,   1,  1928] # EV_ABS / ABS_Y                1928
    - [  0,  11879,   3,  24,    46] # EV_ABS / ABS_PRESSURE           46
    - [  0,  11879,   0,   0,     0] # ------------ SYN_REPORT (0) ------- +0ms
# second device (if any)
- node: /dev/input/event9
evdev: ...
```

Top-level keys are listed below, see the respective subsection for details on each key.

version: int

The file format version. This version is only increased for backwards-incompatible changes. A parser must ignore unknown keys to be forwards-compatible.

ndevices: int

The number of device recordings in this file. Always 1 unless recorded with **--multiple**

libinput: {...}

A dictionary with libinput-specific information.

system: {...}

A dictionary with system information.

devices: {...}
A list of devices containing the description and and events of each device.

libinput

version: string
libinput version

system

Information about the system

os: string Distribution ID and version, see *os-release(5)*

kernel: string
Kernel version, see *uname(1)*

dmi: string
DMI modalias, see */sys/class/dmi/id/modalias*

devices

Information about and events from the recorded device nodes

node: string
the device node recorded

evdev A dictionary with the evdev device information.

hid A list of integers representing the HID report descriptor bytes.

udev A dictionary with the udev device information.

events A list of dictionaries with the recorded events

evdev

name: string
The device name

id: [bustype, vendor, product, version]

The data from the **struct input_id**, bustype, vendor, product, version.

codes: {type: [a, b, c], ...}

All evdev types and codes as nested dictionary. The evdev type is the key, the codes are a list.

absinfo: {code: [min, max, fuzz, flat, resolution], ...}

An array of arrays with 6 decimal elements each, denoting the contents of a **struct input_absinfo**. The first element is the code (e.g. **ABS_X**) in decimal format.

properties: [0, 1, ...]

Array with all **INPUT_PROP_FOO** constants. May be an empty array.

udev

properties: list of strings

A list of udev properties in the **key=value** format. This is not the complete list of properties assigned to the device but a subset that is relevant to libinput. These properties may include properties set on a parent device.

quirks: list of strings

A list of device quirks the **key=value** format.

events

A list of the recorded events. The list contains dictionaries Information about the events. The content is a list of dictionaries, with the string identifying the type of event sequence.

{ evdev: [[sec, usec, type, code, value], ...] }

Each **evdev** dictionary contains the contents of a **struct input_event** in decimal format. The last item in the list is always the **SYN_REPORT** of this event frame. The next event frame starts a new **evdev** dictionary entry in the parent **events** list.

{ hid: hidrawX : [12, 34, 56], ...] }

The **hid** dictionary contains the hid reports in decimal format, with the hidraw node as key. The special key **time** denotes the current time when the report was read from the kernel.

Note that the kernel does not provide timestamps for hidraw events and the timestamps provided are from **clock_gettime(3)**. They may be greater than a subsequent evdev event’s timestamp.

NOTES

This tool records events from the kernel and is independent of libinput. In other words, updating or otherwise changing libinput will not alter the output from this tool. libinput itself does not need to be in use to record events.

LIBINPUT

Part of the **libinput(1)** suite