

NAME

evdev - Generic Linux input driver

SYNOPSIS

```

Section InputDevice
Identifier devname
Driver evdev
Option Device devpath
Option Emulate3Buttons True
Option Emulate3Timeout 50
Option GrabDevice False
...
EndSection

```

DESCRIPTION

evdev is an Xorg input driver for Linux's generic event devices. It therefore supports all input devices that the kernel knows about, including most mice, keyboards, tablets and touchscreens. **evdev** is the default driver on the major Linux distributions.

The **evdev** driver can serve as both a pointer and a keyboard input device. Multiple input devices are supported by multiple instances of this driver, with one `InputDevice` section of your `xorg.conf` for each input device that will use this driver.

It is recommended that **evdev** devices are configured through the **InputClass** directive (refer to `xorg.conf(5)`) instead of manual per-device configuration. Devices configured in the `xorg.conf(5)` are not hot-plug capable.

SUPPORTED HARDWARE

In general, any input device that the kernel has a driver for can be accessed through the **evdev** driver. See the Linux kernel documentation for a complete list.

CONFIGURATION DETAILS

Please refer to `xorg.conf(5)` for general configuration details and for options that can be used with all input drivers. This section only covers configuration details specific to this driver.

The following driver **Options** are supported:

Option ButtonMapping *string*

Sets the button mapping for this device. The mapping is a space-separated list of button mappings that correspond in order to the physical buttons on the device (i.e. the first number is the mapping for button 1, etc.). The default mapping is 1 2 3 ... 32. A mapping of 0 deactivates the button. Multiple buttons can have the same mapping. For example, a left-handed mouse with deactivated scroll-wheel would use a mapping of 3 2 1 0 0. Invalid mappings are ignored and the default mapping is used. Buttons not specified in the user's mapping use the default mapping.

Option Device *string*

Specifies the device through which the device can be accessed. This will generally be of the form `/dev/input/eventX`, where X is some integer. The mapping from device node to hardware is system-dependent. Property: Device Node (read-only).

Option DragLockButtons *L1 B2 L3 B4*

Sets drag lock buttons that simulate holding a button down, so that low dexterity people do not have to hold a button down at the same time they move a mouse cursor. Button numbers occur in pairs, with the lock button number occurring first, followed by the button number that is the target of the lock button. Property: Evdev Drag Lock Buttons.

Option DragLockButtons *M1*

Sets a master drag lock button that acts as a Meta Key indicating that the next button pressed is to be drag locked. Property: Evdev Drag Lock Buttons.

Option Emulate3Buttons *boolean*

Enable/disable the emulation of the third (middle) mouse button for mice which only have two physical buttons. The third button is emulated by pressing both buttons simultaneously. Default: off. Property: Evdev Middle Button Emulation.

Option Emulate3Timeout *integer*

Sets the timeout (in milliseconds) that the driver waits before deciding if two buttons were pressed simultaneously when 3 button emulation is enabled. Default: 50. Property: Evdev Middle Button Timeout.

Option EmulateWheel *boolean*

Enable/disable wheel emulation. Wheel emulation means emulating button press/release events when the mouse is moved while a specific real button is pressed. Wheel button events (typically buttons 4 and 5) are usually used for scrolling. Wheel emulation is useful for getting wheel-like behaviour with trackballs. It can also be useful for mice with 4 or more buttons but no wheel. See the description of the **EmulateWheelButton**, **EmulateWheelInertia**, **EmulateWheelTimeout**, **XAxisMapping**, and **YAxisMapping** options. Default: off. Property: Evdev Wheel Emulation.

Option EmulateWheelButton *integer*

Specifies which button must be held down to enable wheel emulation mode. While this button is down, X and/or Y pointer movement will generate button press/release events as specified for the **XAxisMapping** and **YAxisMapping** settings. If the button is 0 and **EmulateWheel** is on, any motion of the device is converted into wheel events. Default: 4. Property: Evdev Wheel Emulation Button.

Option EmulateWheelInertia *integer*

Specifies how far (in pixels) the pointer must move to generate button press/release events in wheel emulation mode. Default: 10. Property: Evdev Wheel Emulation Inertia.

Option EmulateWheelTimeout *integer*

Specifies the time in milliseconds the **EmulateWheelButton** must be pressed before wheel emulation is started. If the **EmulateWheelButton** is released before this timeout, the original button press/release event is sent. Default: 200. Property: Evdev Wheel Emulation Timeout.

Option EmulateThirdButton *boolean*

Enable third button emulation. Third button emulation emits a right button event (by default) by pressing and holding the first button. The first button must be held down for the configured timeout and must not move more than the configured threshold for the emulation to activate. Otherwise, the first button event is posted as normal. Default: off. Property: Evdev Third Button Emulation.

Option EmulateThirdButtonTimeout *integer*

Specifies the timeout in milliseconds between the initial button press and the generation of the emulated button event. Default: 1000. Property: Evdev Third Button Emulation Timeout.

Option EmulateThirdButtonButton *integer*

Specifies the physical button number to be emitted if third button emulation is triggered. Default: 3. Property: Evdev Third Button Button.

Option EmulateThirdButtonMoveThreshold *integer*

Specifies the maximum move fuzz in device coordinates for third button emulation. If the device moves by more than this threshold before the third button emulation is triggered, the emulation is cancelled and a first button event is generated as normal. Default: 20. Property: Evdev Third Button Emulation Threshold.

Option GrabDevice *boolean*

Force a grab on the event device. Doing so will ensure that no other driver can initialise the same device and it will also stop the device from sending events to `/dev/kbd` or `/dev/input/mice`. Events from this device will not be sent to virtual devices (e.g. rfbkill or the Macintosh mouse button emulation). Default: disabled.

Option InvertX *Bool***Option InvertY** *Bool*

Invert the given axis. Default: off. Property: Evdev Axis Inversion.

Option IgnoreRelativeAxes *Bool***Option IgnoreAbsoluteAxes** *Bool*

Ignore the specified type of axis. Default: unset. The X server cannot deal with devices that have both relative and absolute axes. Evdev tries to guess which axes to ignore given the device type and disables absolute axes for mice and relative axes for tablets, touchscreens and touchpad. These options allow to forcibly disable an axis type. Mouse wheel axes are exempt and will work even if relative axes are ignored. No property, this configuration must be set in the configuration.

If either option is set to False, the driver will not ignore the specified axes regardless of the presence of other axes. This may trigger buggy behavior and events from this axis are always forwarded. Users are discouraged from setting this option.

Option Calibration *min-x max-x min-y max-y*

Calibrates the X and Y axes for devices that need to scale to a different coordinate system than reported to the X server. This feature is required for devices that need to scale to a different coordinate system than originally reported by the kernel (e.g. touchscreens). The scaling to the custom coordinate system is done in-driver and the X server is unaware of the transformation. Property: Evdev Axis Calibration.

Option Mode Relative|Absolute

Sets the mode of the device if device has absolute axes. The default value for touchpads is relative, for other absolute. This option has no effect on devices without absolute axes.

Option SwapAxes *Bool*

Swap x/y axes. Default: off. Property: Evdev Axes Swap.

Option XAxisMapping *N1 N2*

Specifies which buttons are mapped to motion in the X direction in wheel emulation mode. Button number *N1* is mapped to the negative X axis motion and button number *N2* is mapped to the positive X axis motion. Default: no mapping. Property: Evdev Wheel Emulation Axes.

Option YAxisMapping *N1 N2*

Specifies which buttons are mapped to motion in the Y direction in wheel emulation mode. Button number *N1* is mapped to the negative Y axis motion and button number *N2* is mapped to the positive Y axis motion. Default: 4 5. Property: Evdev Wheel Emulation Axes.

Option TypeName *type*

Specify the X Input 1.x type (see `XListInputDevices(3)`) There is rarely a need to use this option, evdev will guess the device type based on the device's capabilities. This option is provided for devices that need quirks.

Option VertScrollDelta *integer*

The amount of motion considered one unit of scrolling vertically. Default: 1. Property: Evdev Scrolling Distance.

Option HorizScrollDelta *integer*

The amount of motion considered one unit of scrolling horizontally. Default: 1. Property: Evdev Scrolling Distance.

Option DialDelta *integer*

The amount of motion considered one unit of turning the dial. Default: 1. Property: Evdev Scrolling Distance.

SUPPORTED PROPERTIES

The following properties are provided by the **evdev** driver.

Evdev Axis Calibration

4 32-bit values, order min-x, max-x, min-y, max-y or 0 values to disable in-driver axis calibration.

Evdev Axis Inversion

2 boolean values (8 bit, 0 or 1), order X, Y. 1 inverts the axis.

Evdev Axes Swap

1 boolean value (8 bit, 0 or 1). 1 swaps x/y axes.

Evdev Drag Lock Buttons

8-bit. Either 1 value or pairs of values. Value range 0-32, 0 disables a value.

Evdev Middle Button Emulation

1 boolean value (8 bit, 0 or 1).

Evdev Middle Button Timeout

1 16-bit positive value.

Evdev Wheel Emulation

1 boolean value (8 bit, 0 or 1).

Evdev Wheel Emulation Axes

4 8-bit values, order X up, X down, Y up, Y down. 0 disables a value.

Evdev Wheel Emulation Button

1 8-bit value, allowed range 0-32, 0 disables the button.

Evdev Wheel Emulation Inertia

1 16-bit positive value.

Evdev Wheel Emulation Timeout

1 16-bit positive value.

Evdev Scrolling Distance

3 32-bit values: vertical, horizontal and dial.

AUTHORS

Kristian Høgsberg, Peter Hutterer

SEE ALSO

[Xorg\(1\)](#), [xorg.conf\(5\)](#), [Xserver\(1\)](#), [X\(7\)](#)