NAME

mouse − Xorg mouse input driver

SYNOPSIS

Section "InputDevice"
Identifier "*idevname***"**
Driver "mouse"
Option "Protocol" "*protoname***"**
Option "Device" "*devpath***"**
...
EndSection

DESCRIPTION

mouse is an Xorg input driver for mice. The driver supports most available mouse types and interfaces, though the level of support for types of mice depends on the OS.

The **mouse** driver functions as a pointer input device. Multiple mice are supported by multiple instances of this driver.

SUPPORTED HARDWARE

USB mouse

USB (Universal Serial Bus) ports are present on most modern computers. Several devices can be plugged into this bus, including mice and keyboards. Support for USB mice is platform specific.

PS/2 mouse

The PS/2 mouse is an intelligent device and may have more than three buttons and a wheel or a roller. The PS/2 mouse is usually compatible with the original PS/2 mouse from IBM immediately after power up. The PS/2 mouse with additional features requires a specialized initialization procedure to enable these features. Without proper initialization, it behaves as though it were an ordinary two or three button mouse.

Serial mouse

There have been numerous serial mouse models from a number of manufacturers. Despite the wide range of variations, there have been relatively few protocols (data format) with which the serial mouse talks to the host computer.

The modern serial mouse conforms to the PnP COM device specification so that the host computer can automatically detect the mouse and load an appropriate driver. This driver supports this specification and can detect popular PnP serial mouse models on most platforms.

Bus mouse

The bus mouse connects to a dedicated interface card in an expansion slot. Some older video cards, notably those from ATI, and integrated I/O cards may also have a bus mouse connector.

The interface type of the mouse can be determined by looking at the connector of the mouse. USB mice have a thin rectangular connector. PS/2 mice are equipped with a small, round DIN 6-pin connector. Serial mouse have a D-Sub female 9- or 25-pin connector. Bus mice have either a D-Sub male 9-pin connector or a round DIN 9-pin connector. Some mice come with adapters with which the connector can be converted to another. If you are to use such an adapter, remember that the connector at the very end of the mouse/adapter pair is what matters.

CONFIGURATION DETAILS

Depending on the X server version in use, input device options may be set in either a xorg.conf file, an xorg.conf.d snippet or in the configuration files read by the Hardware Abstraction Layer (HAL) daemon, hald(1).

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 driver can auto-detect the mouse type on some platforms. On some platforms this is limited to plug and play serial mice, and on some the auto-detection works for any mouse that the OS's kernel driver

supports. On others, it is always necessary to specify the mouse protocol in the config file. The *README* document provided with this driver contains some detailed information about this.

The following driver **Options** are supported:

Option "Protocol" "*string***"**
Specify the mouse protocol. Valid protocol types include:

Auto, Microsoft, MouseSystems, MMSeries, Logitech, MouseMan, MMHitTab, GlidePoint, IntelliMouse, ThinkingMouse, ValuMouseScroll, AceCad, PS/2, ImPS/2, ExplorerPS/2, ThinkingMousePS/2, MouseManPlusPS/2, GlidePointPS/2, NetMousePS/2, NetScrollPS/2, BusMouse, SysMouse, WSMouse, USB, VUID, Xqueue.

Not all protocols are supported on all platforms. The "Auto" protocol specifies that protocol auto-detection should be attempted. The default protocol setting is platform-specific.

Option "Device" "*string***"**
Specifies the device through which the mouse can be accessed. A common setting is "/dev/mouse", which is often a symbolic link to the real device. This option is mandatory, and there is no default setting. The driver may however attempt to probe some default devices if this option is missing. Property: "Device Node" (read-only).

Option "Buttons" "*integer***"**
Specifies the number of mouse buttons. In cases where the number of buttons cannot be auto-detected, the default value is 3. The maximum number is 24.

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: on, until a press of a physical button 3 is detected. Property: "Mouse Middle Button Emulation"

Option "Emulate3Timeout" "*integer***"**
Sets the timeout (in milliseconds) that the driver waits before deciding if two buttons where pressed "simultaneously" when 3 button emulation is enabled. Default: 50. Property: "Mouse Middle Button Timeout"

Option "ChordMiddle" "*boolean***"**
Enable/disable handling of mice that send left+right events when the middle button is used. Default: off.

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**, **XAxisMapping**, and **YAxisMapping** options below. Default: off.

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 set to 0, no button is required and any motion of the device is converted into wheel events. Default: 4.

Option "EmulateWheelInertia" "*integer***"**
Specifies how far (in pixels) the pointer must move to generate button press/release events in wheel emulation mode. Default: 10.

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.

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.

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: no mapping.

Option "ZAxisMapping" "X"

Option "ZAxisMapping" "Y"

Option "ZAxisMapping" "*N1 N2***"**

Option "ZAxisMapping" "*N1 N2 N3 N4***"**
Set the mapping for the Z axis (wheel) motion to buttons or another axis (**X** or **Y**). Button number *N1* is mapped to the negative Z axis motion and button number *N2* is mapped to the positive Z axis motion. For mice with two wheels, four button numbers can be specified, with the negative and positive motion of the second wheel mapped respectively to buttons number *N3* and *N4*. Note that the protocols for mice with one and two wheels can be different and the driver may not be able to autodetect it. Default: "4 5".

Option "ButtonMapping" "*N1 N2 [...]***"**
Specifies how physical mouse buttons are mapped to logical buttons. Physical button 1 is mapped to logical button *N1*, physical button 2 to *N2*, and so forth. This enables the use of physical buttons that are obscured by *ZAxisMapping*. Default: "1 2 3 8 9 10 ...".

Option "FlipXY" "*boolean***"**
Enable/disable swapping the X and Y axes. This transformation is applied after the **InvX**, **InvY** and **AngleOffset** transformations. Default: off.

Option "InvX" "*boolean***"**
Invert the X axis. Default: off.

Option "InvY" "*boolean***"**
Invert the Y axis. Default: off.

Option "AngleOffset" "*integer***"**
Specify a clockwise angular offset (in degrees) to apply to the pointer motion. This transformation is applied before the **FlipXY**, **InvX** and **InvY** transformations. Default: 0.

Option "SampleRate" "*integer***"**
Sets the number of motion/button events the mouse sends per second. Setting this is only supported for some mice, including some Logitech mice and some PS/2 mice on some platforms. Default: whatever the mouse is already set to.

Option "Resolution" "*integer***"**
Sets the resolution of the device in counts per inch. Setting this is only supported for some mice, including some PS/2 mice on some platforms. Default: whatever the mouse is already set to.

Option "Sensitivity" " *float***"**
Mouse movements are multiplied by this float before being processed. Use this mechanism to slow down high resolution mice. Because values bigger than 1.0 will result in not all pixels on the screen being accessible, you should better use mouse acceleration (see **man xset**) for speeding up low resolution mice. Default: 1.0

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.

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".

Option "ClearDTR" "*boolean***"**
Enable/disable clearing the DTR line on the serial port used by the mouse. Some dual-protocol mice require the DTR line to be cleared to operate in the non-default protocol. This option is for serial mice only and is handled by the X server. Default: off.

Option "ClearRTS" "*boolean***"**
Enable/disable clearing the RTS line on the serial port used by the mouse. Some dual-protocol mice require the RTS line to be cleared to operate in the non-default protocol. This option is for serial mice only and is handled by the X server. Default: off.

Option "BaudRate" "*integer***"**
Set the baud rate to use for communicating with a serial mouse. This option should rarely be required because the default is correct for almost all situations. Valid values include: 300, 1200, 2400, 4800, 9600, 19200. Default: 1200.

There are some other options that may be used to control various parameters for serial port communication, but they are not documented here because the driver sets them correctly for each mouse protocol type.

SEE ALSO

Xorg(1), xorg.conf(5), Xserver(1), X(7), README.mouse.

hal(7), hald(8), fdi(5).