Mouse Support in XFree86

Kazutaka Yokota

9 February 2000

1. Introduction

This document describes mouse support in XFree86 4.1.0.

Mouse configuration has often been mysterious task for novice users. However, once you learn several basics, it is straightforward to write the mouse `"InputDevice"` section in the `XF86Config` file by hand.

2. Supported Hardware

The XFree86 X server supports three classes of mice: serial, bus and PS/2 mice.

Serial mouse

The serial mouse has been the most popular pointing device for PCs. There have been numerous serial mouse models from a number of manufactures. 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. The XFree86 X server 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 video cards, notably those from ATI, and integrated I/O cards may also have a bus mouse connector. Some bus mice are known as 'InPort mouse'.

Note that some mouse manufactures have sold a package including a serial mouse and a serial interface card. Don't confuse this type of products with the genuine bus mouse.

PS/2 mouse

They are sometimes called 'Mouse-port mouse'. The PS/2 mouse is becoming increasingly common and popular.

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.

USB mouse

USB (Universal Serial Bus) ports are present on most modern computers. Several devices can be plugged into this bus, including mices and keyboards.

The XFree86 server includes support for USB mices on some systems.

Many mice nowadays can be used both as a serial mouse and as a PS/2 mouse. They has a logic to distinguish which interface it is connected to. However, the mouse which is not marketed as compatible with both serial and PS/2 mouse interface lacks this logic and cannot be used in such a way, even if you can find an appropriate adapter with which you can connect the PS/2 mouse to a serial port or visa versa.

XFree86 supports the mouse with a wheel, a roller or a knob. Its action is detected as the Z (third) axis motion of the mouse. As the X server or clients normally do not use the Z axis movement of the pointing device, a configuration option, `"ZAxisMapping"`, is provided to assign the Z axis movement to another axis or a pair of buttons (see below).

3. OS Support for Mice

3.1 Summary of Supported Mouse Protocol Types

Protocol Types

OS platforms	serial protocols	PnP serial "Auto"	BusMouse protocol "BusMouse"	PS/2 protocol "PS/2"	Extended PS/2 protocols "xxxPS/2"	USB
BSD/OS	Ok	?	?	?	?	?
FreeBSD	Ok	Ok	Ok	Ok	SP*1	SP*1
FreeBSD(98)	Ok	?	Ok	NA	NA	?
Interactive Unix	Ok	NA	?*1	?*1	NA	?
Linux	Ok	Ok	Ok	Ok	Ok	?
Linux/98	Ok	?	Ok	NA	NA	?
LynxOS	Ok	NA	Ok	Ok	NA	?
NetBSD	Ok	Ok	Ok	SP*1	SP*1	SP*1
NetBSD/pc98	Ok	?	Ok	NA	NA	NA
OpenBSD	Ok	Ok	Ok	Ok*1	Ok*1	Ok*1
OS/2	SP*2	SP*2	SP*2	SP*2	SP*2	?
SCO	Ok	?	SP*1	SP*1	NA	?
Solaris 2.x	Ok	NA*1	?*1	Ok	Ok	?
SVR4	Ok	NA*1	SP*1	SP*1	NA	?
PANIX	Ok	?	SP*1	SP*1	NA	?

Ok: support is available, NA: not available, ?: untested or unknown.
SP: support is available in a different form

*1 Refer to the following sections for details.
*2 XFree86/OS2 will support any type of mouse that the OS supports, whether it is serial, bus mouse, or PnP type.

3.2 BSD/OS

No testing has been done with BSD/OS.

3.3 FreeBSD

FreeBSD supports the `"SysMouse"` protocol which must be specified when the `moused` daemon is running in versions 2.2.1 or later.

When running the `moused`daemon, you must always specify the `/dev/sysmouse` device and the `"SysMouse"` protocol to the X server, regardless of the actual type of your mouse.

FreeBSD versions 2.2.6 or later include the kernel-level support for extended PS/2 mouse protocols and there is no need to specify the exact protocol name to the X server. Instead specify the

"PS/2" or "Auto" protocol and the X server will automatically make use of the kernel-level support.

In fact, "Auto" protocol support is really efficient in these versions. You may always specify "Auto" to any mouse, serial, bus or PS/2, unless the mouse is an old serial model which doesn't support PnP.

FreeBSD versions 2.2.5 or earlier do not support extended PS/2 mouse protocols ("xxxPS/2"). Always specify the "PS/2" protocol for any PS/2 mouse in these versions regardless of the brand of the mouse.

FreeBSD versions 3.1 or later have support for USB mice. Specify the "Auto" protocol for the /dev/ums0 device. (If the moused daemon is running for the USB mouse, you must use /dev/sysmouse instead of /dev/ums0 as explained above.) See the *ums(4)* manual page for details.

3.4 FreeBSD(98)

The PS/2 mouse is not supported.

3.5 Interactive Unix

The PnP serial mouse support (the "Auto" protocol) is not supported for the moment.

The bus mouse and PS/2 mouse should be supported by using the appropriate device drivers. Use /dev/mouse for the "BusMouse" protocol and /dev/kdmouse for the "PS/2" protocol. These protocols are untested but may work. Please send success/failure reports to <michael.rohleder@stadt-frankfurt.de>.

3.6 Linux

All protocol types should work.

3.7 Linux/98

The PS/2 mouse is not supported.

3.8 LynxOS

The PnP serial mouse support (the "Auto" protocol) is disabled in LynxOS, because of limited TTY device driver functionality.

3.9 NetBSD

NetBSD 1.3.x and former does not support extended PS/2 mouse protocols ("xxxPS/2"). The PS/2 mouse device driver /dev/pms emulates the bus mouse. Therefore, you should always specify the "BusMouse" protocol for any PS/2 mouse regardless of the brand of the mouse.

The "wsmouse" protocol introduced in NetBSD 1.4 along with the wscons console driver is supported. You need to run binaries compiled on NetBSD 1.4 to have support for it though. Use "/dev/wsmouse0" for the device. Refer to the *wsmouse(4)* manual page for kernel configuration informations.

This driver also provides support for USB mices. See the *ums(4)* manual page for details.

3.10 NetBSD/pc98

The PS/2 mouse is not supported.

3.11 OpenBSD

The raw PS/2 mouse device driver /dev/psm0 uses the raw PS/2 mouse protocol.

OpenBSD 2.2 and earlier does not support extended PS/2 mouse protocols (`"xxxPS/2"`). Therefore, you should specify the `"PS/2"` protocol for any PS/2 mouse regardless of the brand of the mouse.

OpenBSD 2.3 and later support all extended PS/2 mouse protocols. You can select the `"Auto"` protocol for PnP PS/2 mice or any specific extended (`"xxxPS/2"`) protocol for non PnP mice.

There is also a cooked PS/2 mouse device driver `/dev/pms0` which emulates the bus mouse. Specify the `"BusMouse"` protocol for any PS/2 mouse regardless of the brand of the mouse when using this device.

XFree86 3.3.6 support USB mices on OpenBSD 2.6 and later though the generic Human Interface Device (hid) `/dev/uhid*`. Select the `"usb"` protocol and the `/dev/uhid*` instance corresponding to your mouse as the device name.

3.12 OS/2

XFree86/OS2 always uses the native mouse driver of the operating system and will support any type of pointer that the OS supports, whether it is serial, bus mouse, or PnP type. If the mouse works under Presentation Manager, it will also work under XFree86/OS2.

Always specify `"OSMouse"` as the protocol type.

3.13 SCO

The bus and PS/2 mouse are supported with the `"OSMouse"` protocol type.

The `"OSMouse"` may also be used with the serial mouse.

3.14 Solaris

Testing has been done with Solaris 2.5.1 and 2.6. Logitech and Microsoft bus mice have not been tested, but might work with the `/dev/logi` and `/dev/msm` devices. Standard 2 and 3 button PS/2 mice work with the `"PS/2"` protocol type and the `/dev/kdmouse` device. The PnP serial mouse support (the `"Auto"` protocol) has been tested and does not work.

3.15 SVR4

The bus and PS/2 mouse may be supported with the `"Xqueue"` protocol type.

The `"Xqueue"` may also be used with the serial mouse.

The PnP serial mouse support (the `"Auto"` protocol) is not tested.

3.16 PANIX

The PC/AT version of PANIX supports the bus and PS/2 mouse with the `"Xqueue"` protocol type. The PC-98 version of PANIX supports the bus mouse with the `"Xqueue"` protocol type.

4. Configuring Your Mouse

Before using the `xf86config` program to set up mouse configuration, you must identify the interface type, the device name and the protocol type of your mouse. Blindly trying every possible combination of mouse settings will lead you nowhere.

The first thing you need to know is the interface type of the mouse you are going to use. It can be determined by looking at the connector of the mouse. The serial mouse has a D-Sub female 9- or 25-pin connector. The bus mice have either a D-Sub male 9-pin connector or a round DIN 9-pin connector. The PS/2 mouse is equipped with a small, round DIN 6-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.

The next thing to decide is a device node to use for the given interface. For the bus and PS/2 mice, there is little choice; your OS most possibly offers just one device node each for the bus mouse and PS/2 mouse. There may be more than one serial port to which the serial mouse can be attached.

The next step is to guess the appropriate protocol type for the mouse. The X server may be able to select a protocol type for the given mouse automatically in some cases. Otherwise, the user has to choose one manually. Follow the guidelines below.

Bus mouse
: The bus and InPort mice always use `"BusMouse"` protocol regardless of the brand of the mouse.

  Some OSs may allow you to specify `"Auto"` as the protocol type for the bus mouse.

PS/2 mouse
: The `"PS/2"` protocol should always be tried first for the PS/2 mouse regardless of the brand of the mouse. Any PS/2 mouse should work with this protocol type, although wheels and other additional features are unavailable in the X server.

  After verifying the mouse works with this protocol, you may choose to specify one of `"xxxPS/2"` protocols so that extra features are made available in the X server. However, support for these PS/2 mice assumes certain behavior of the underlying OS and may not always work as expected. Support for some PS/2 mouse models may be disabled all together for some OS platforms for this reason.

  Some OSs may allow you to specify `"Auto"` as the protocol type for the PS/2 mouse and the X server will automatically adjust itself.

Serial mouse
: The XFree86 server supports a wide range of mice, both old and new. If your mouse is of a relatively new model, it may conform to the PnP COM device specification and the X server may be able to detect an appropriate protocol type for the mouse automatically.

  Specify `"Auto"` as the protocol type and start the X server. If the mouse is not a PnP mouse, or the X server cannot determine a suitable protocol type, the server will print the following error message and abort.

  ```
  <mousename>: cannot determine the mouse protocol
  ```

  If the X server generates the above error message, you need to manually specify a protocol type for your mouse. Choose one from the following list:

  - `GlidePoint`
  - `IntelliMouse`
  - `Logictech`
  - `Microsoft`
  - `MMHittab`
  - `MMSeries`
  - `MouseMan`
  - `MouseSystems`
  - `ThinkingMouse`

  When you choose, keep in mind the following rule of thumb:

1. `"Logitech"` protocol is for old serial mouse models from Logitech. Modern Logitech mice use either `"MouseMan"` or `"Microsoft"` protocol.
2. Most 2-button serial mice support the `"Microsoft"` protocol.
3. 3-button serial mice may work with the `"Mousesystems"` protocol. If it doesn't, it may work instead with the `"Microsoft"` protocol although the third (middle) button won't function. 3-button serial mice may also work with the `"Mouseman"` protocol under which the third button may function as expected.
4. 3-button serial mice may have a small switch at the bottom of the mouse to choose between "MS" and "PC", or "2" and "3". "MS" or "2" usually mean the `"Microsoft"` protocol. "PC" or "3" will choose the `"MouseSystems"` protocol.
5. If the serial mouse has a roller or a wheel, it may be compatible with the `"IntelliMouse"` protocol.
6. If the serial mouse has a roller or a wheel and it doesn't work with the `"IntelliMouse"` protocol, you have to use it as a regular 2- or 3-button serial mouse.

If the `"Auto"` protocol is specified and the mouse seems working, but you find that not all features of the mouse is available, that is because the X server does not have native support for that model of mouse and is using a "compatible" protocol according to PnP information.

If you suspect this is the case with your mouse, please send a report to <XFree86@XFree86.Org>.

USB mouse
: If your mouse is connected to the USB port, it can either be supported by the `"Auto"` protocol, or by an OS-specific protocol (see below), or as a generic Human Interface Device by the `"usb"` protocol.

Standardized protocols
: Mouse device drivers in your OS may use the standardized protocol regardless of the model or the class of the mouse. For example, SVR4 systems may support `"Xqueue"` protocol. In FreeBSD the system mouse device `/dev/sysmouse` uses the `"SysMouse"` protocol. Please refer to the OS support section of this file for more information.

5. XF86Config Options

The old `Pointer` section has been replaced by a more general `InputDevice` section. The following is a minimal example of an `InputDevice` section for a mouse:

```
Section "InputDevice"
        Identifier     "Mouse 1"
        Driver         "mouse"
        Option         "Device"       "/dev/mouse"
        Option         "Protocol"     "Auto"
EndSection
```

The `mouse` driver supports the following config file options:

5.1 Buttons

This option tells the X server the number of buttons on the mouse. Currently there is no reliable way to automatically detect the correct number. This option is the only means for the X server to obtain it. The default value is three.

Note that if you intend to assign Z axis movement to button events using the `ZAxisMapping` option below, you need to take account of those buttons into `N` too.

```
Option "Buttons"      "N"
```

5.2 ZAxisMappping

This option maps the Z axis (wheel) motion to buttons or to another axis.

```
Option "ZAxisMapping" "X"
Option "ZAxisMapping" "Y"
Option "ZAxisMapping" "N1 N2"
Option "ZAxisMapping" "N1 N2 N3 N4"
```

The first example will map the Z axis motion to the X axis motion. Whenever the user moves the wheel/roller, its movement is reported as the X axis motion. When the wheel/roller stays still, the real X axis motion is reported as is. The third example will map negative Z axis motion to the button `N1` and positive Z axis motion to the button `N2`. If this option is used and the buttons `N1` or `N2` actually exists in the mouse, their actions won't be detected by the X server.

The last example is useful for the mouse with two wheels of which the second wheel is used to generate horizontal scroll action, and the mouse which has a knob or a stick which can detect the horizontal force applied by the user. The motion of the second wheel will be mapped to the buttons `N3`, for the negative direction, and `N4`, for the positive direction. If the buttons `N3` and `N4` actually exist in this mouse, their actions won't be detected by the X server.

NOTE #1: horizontal movement may not always be detected by the current version of the XFree86 X servers, because there appears to be no accepted standard as to how the horizontal direction is encoded in mouse data.

NOTE #2: Some mice think left is the negative horizontal direction, others may think otherwise. Moreover, there are some mice whose two wheels are both mounted vertically, and the direction of the second vertical wheel does not match the first one's.

Currently this option can not be set in the `XF86Setup` program. You need to edit the `XF86Config` file by hand to add this option.

5.3 Resolution

The following option will set the mouse device resolution to `N` counts per inch, if possible:

```
Option "Resolution"   "N"
```

Not all mice and OSs can support this option. This option can be set in the `XF86Setup` program.

6. Mouse Gallery

In all of the examples below, it is assumed that `/dev/mouse` is a link to the appropriate serial port or PS/2 mouse device.

6.1 MS IntelliMouse (serial, PS/2)

This mouse has a wheel which also acts as the button 2 (middle button). The wheel movement is recognized as the Z axis motion. This behavior is not compatible with XFree86 versions prior to 3.3.2, but is more consistent with the support for other mice with wheels or rollers. If you want to make the wheel behave like before, you can use the `"ZAxisMapping"` option as described above.

IntelliMouse supports the PnP COM device specification.

To use this mouse as a serial device:

```
Option "Protocol"     "Auto"
```

or:

```
Option "Protocol"    "IntelliMouse"
```

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"    "IMPS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the wheel won't work in this case):

```
Option "Protocol"    "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"    "Auto"
```

6.2 MS IntelliMouse Explorer (PS/2, USB)

This mouse has a wheel which also acts as the button 2 (middle button). There are two side buttons; they are recognized as the buttons 4 and 5. The wheel movement is recognized as the Z axis motion.

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"    "ExplorerPS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the wheel and the side buttons won't work in this case):

```
Option "Protocol"    "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"    "Auto"
```

To use this mouse as the USB device and the OS supports the generic HID protocol:

```
Option "Protocol"    "usb"
```

To use this mouse as the USB device and the OS supports automatic mouse detection:

```
Option "Protocol"    "Auto"
```

6.3 Kensington Thinking Mouse (serial, PS/2)

This mouse has four buttons. Thinking Mouse supports the PnP COM device specification.

To use this mouse as a serial device:

```
Option "Protocol"    "Auto"
```

or:

```
Option "Protocol"    "ThinkingMouse"
```

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"    "ThinkingMousePS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the third and the fourth buttons act as though they were the first and the second buttons):

```
Option "Protocol"    "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"    "Auto"
```

6.4 Genius NetScroll (PS/2)

This mouse has four buttons and a roller. The roller movement is recognized as the Z axis motion.

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"    "NetScrollPS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the roller and the fourth button won't work):

```
Option "Protocol"    "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"    "Auto"
```

6.5 Genius NetMouse and NetMouse Pro (serial, PS/2)

These mice have a "magic button" which is used like a wheel or a roller. The "magic button" action is recognized as the Z axis motion. NetMouse Pro is identical to NetMouse except that it has the third button on the left hand side.

NetMouse and NetMouse Pro support the PnP COM device specification. When used as a serial mouse, they are compatible with MS IntelliMouse.

To use these mice as a serial device:

```
Option "Protocol"    "Auto"
```

or:

```
Option "Protocol"    "IntelliMouse"
```

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"    "NetMousePS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the "magic button" and the third button won't work):

```
Option "Protocol"    "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"    "Auto"
```

6.6 Genius NetScroll Optical (PS/2, USB)

This mouse has a wheel which also acts as the button 2 (middle button), and two side buttons which are recognized as the buttons 4 and 5. It is compatible with NetMouse and NetMouse Pro.

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"    "NetMousePS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the wheel and the side buttons won't work):

```
Option "Protocol"    "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"    "Auto"
```

To use this mouse as the USB device and the OS supports the generic HID protocol:

```
Option "Protocol"    "usb"
```

To use this mouse as the USB device and the OS supports automatic mouse detection:

```
Option "Protocol" "Auto"
```

6.7 ALPS GlidePoint (serial, PS/2)

The serial version of this pad device has been supported since XFree86 3.2. 'Tapping' action is interpreted as the fourth button press. (IMHO, the fourth button of GlidePoint should always be mapped to the first button in order to make this pad behave like the other pad products.)

To use this pad as a serial device:

```
Option "Protocol" "GlidePoint"
```

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol" "GlidePointPS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization:

```
Option "Protocol" "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol" "Auto"
```

6.8 ASCII MieMouse (serial, PS/2)

This mouse appears to be OEM from Genius. Although its shape is quite different, it works like Genius NetMouse Pro. This mouse has a "knob" which is used like a wheel or a roller. The "knob" action is recognized as the Z axis motion.

MieMouse supports the PnP COM device specification. When used as a serial mouse, it is compatible with MS IntelliMouse.

To use this mouse as a serial device:

```
Option "Protocol" "Auto"
```

or:

```
Option "Protocol" "IntelliMouse"
```

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol" "NetMousePS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the knob and the third button won't work):

```
Option "Protocol" "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol" "Auto"
```

6.9 Logitech MouseMan+ and FirstMouse+ (serial, PS/2)

MouseMan+ has two buttons on top, one side button and a roller. FirstMouse+ has two buttons and a roller. The roller movement is recognized as the Z axis motion. The roller also acts as the third button. The side button is recognized as the fourth button.

MouseMan+ and FirstMouse+ support the PnP COM device specification. They have MS IntelliMouse compatible mode when used as a serial mouse.

To use these mice as a serial device:

```
Option "Protocol" "Auto"
```

or:

```
Option "Protocol"     "IntelliMouse"
```

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"     "MouseManPlusPS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the wheel and the fourth button won't work):

```
Option "Protocol"     "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"     "Auto"
```

6.10 IBM ScrollPoint (PS/2)

ScrollPoint has a "stick" in between the two buttons. This "stick" is the same as the stick-shaped pointing device often found on notebook computers, on which you move the mouse cursor by pushing the stick. The stick movement is recognized as the Z axis motion. You can push the stick to right and left, as well as forward and backward. Give four numbers to `ZAxisMapping` option to map movement along all these four directions to button actions.

This mouse is compatible with Logitech MouseMan+. To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"     "MouseManPlusPS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the stick won't work):

```
Option "Protocol"     "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"     "Auto"
```

6.11 8D ScrollMouse (serial, PS/2)

ScrollMouse, also known as GyroMouse, has a "stick" similar to IBM ScrollPoint. The stick movement is recognized as the Z axis motion. You can push the stick to right and left, as well as forward and backward. Give four numbers to `ZAxisMapping` option to map movement along all these four directions to button actions.

ScrollMouse supports the PnP COM device specification. When used as a serial mouse, it is compatible with MS IntelliMouse.

To use this mouse as a serial device:

```
Option "Protocol"     "Auto"
```

or:

```
Option "Protocol"     "IntelliMouse"
```

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"     "IMPS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the stick won't work):

```
Option "Protocol"     "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"     "Auto"
```

6.12 A4 Tech 4D mice (serial, PS/2, USB)

A4 Tech produces quit a number of mice with one or two wheels. Their mice may have 2, 3, or 4 buttons. The wheels movement is recognized as the Z axis motion. Give four numbers to `ZAxisMapping` option to map movement of both wheels to button actions.

4D mice support the PnP COM device specification. When used as a serial mouse, it is compatible with MS IntelliMouse.

To use this mouse as a serial device:

```
Option "Protocol"     "Auto"
```

or:

```
Option "Protocol"     "IntelliMouse"
```

To use this mouse as the PS/2 device and the OS supports PS/2 mouse initialization:

```
Option "Protocol"     "IMPS/2"
```

To use this mouse as the PS/2 device but the OS does not support PS/2 mouse initialization (the wheels won't work):

```
Option "Protocol"     "PS/2"
```

To use this mouse as the PS/2 device and the OS supports automatic PS/2 mouse detection:

```
Option "Protocol"     "Auto"
```

To use this mouse as the USB device and the OS supports the generic HID protocol:

```
Option "Protocol"     "usb"
```

To use this mouse as the USB device and the OS supports automatic mouse detection:

```
Option "Protocol"     "Auto"
```

7. Configuration Examples

This section shows some example `InputDevice` section for popular mice. All the examples assume that the mouse is connected to the PS/2 mouse port, and the OS supports the PS/2 mouse initialization. It is also assumed that `/dev/mouse` is a link to the PS/2 mouse port.

Logitech MouseMan+ has 4 buttons and a wheel. The following example makes the wheel movement available as the button 5 and 6.

```
Section "InputDevice"
        Identifier     "MouseMan+"
        Driver         "mouse"
        Option         "Device"       "/dev/mouse"
        Option         "Protocol"     "MouseManPlusPS/2"
        Option         "Buttons"      "6"
        Option         "ZAxisMapping" "5 6"
EndSection
```

You can change button number assignment using the `xmodmap` command AFTER you start the X server with the above configuration. You may not like to use the wheel as the button 2 and rather want the side button (button 4) act like the button 2. You may also want to map the wheel movement to the button 4 and 5. This can be done by the following command:

```
xmodmap -e "pointer = 1 6 3 2 4 5"
```

After this command is run, the correspondence between the buttons and button numbers will be as shown in the following table.

```
Physical Buttons       Reported as:
------------------------------------
1 Left Button            Button 1
2 Wheel Button           Button 6
3 Right Button           Button 3
4 Side Button            Button 2
5 Wheel Negative Move    Button 4
6 Wheel Positive Move    Button 5
```

For the MS IntelliMouse Explorer which as a wheel and 5 buttons, you may have the following `InputDevice` section.

```
Section "InputDevice"
       Identifier     "IntelliMouse Explorer"
       Driver         "mouse"
       Option         "Device"       "/dev/mouse"
       Option         "Protocol"     "ExplorerPS/2"
       Option         "Buttons"      "7"
       Option         "ZAxisMapping" "6 7"
EndSection
```

The IntelliMouse Explorer has 5 buttons, thus, you should give "7" to the `Buttons` option if you want to map the wheel movement to buttons (6 and 7). With this configuration, the correspondence between the buttons and button numbers will be as follows:

```
Physical Buttons       Reported as:
------------------------------------
1 Left Button            Button 1
2 Wheel Button           Button 2
3 Right Button           Button 3
4 Side Button 1                  Button 4
5 Side Button 2                  Button 5
6 Wheel Negative Move    Button 6
7 Wheel Positive Move    Button 7
```

You can change button number assignment using `xmodmap` AFTER you started the X server with the above configuration.

```
xmodmap -e "pointer = 1 2 3 4 7 5 6"
```

The above command will moves the side button 2 to the button 7 and make the wheel movement reported as the button 5 and 6. See the table below.

```
Physical Buttons       Reported as:
------------------------------------
1 Left Button            Button 1
2 Wheel Button           Button 2
3 Right Button           Button 3
4 Side Button 1                  Button 4
5 Side Button 2                  Button 7
6 Wheel Negative Move    Button 5
7 Wheel Positive Move    Button 6
```

For the A4 Tech WinEasy mouse which has two wheels and 3 buttons, you may have the following `InputDevice` section.

```
Section "InputDevice"
       Identifier     "WinEasy"
       Driver         "mouse"
       Option         "Device"       "/dev/mouse"
       Option         "Protocol"     "IMPS/2"
       Option         "Buttons"      "7"
       Option         "ZAxisMapping" "4 5 6 7"
EndSection
```

The movement of the first wheel is mapped to the button 4 and 5. The second wheel's movement will be reported as the buttons 6 and 7.

CONTENTS

`$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/mouse.sgml,v 1.11 2000/03/01 00:25:23 dawes Exp $`