Wacom−Tablet−HOWTO

Table of Contents

Wacom Tablet HOWTO..........1
Stefan Runkel <runkel@runkeledv.de>..........1
1. Copyright..........1
2. Introduction..........1
3. Requirements..........1
4. Tablets as Mouse−Replacement for the Linux−Console..........1
5. Tablets with XFree 3 and XFree 4..........1
6. Changing configuration from within a running X−Server..........2
7. XFree FAQ − Problems And Questions..........2
8. Gpm FAQ − Problems And Questions..........2
9. Further Information / Used Documents..........2
1. Copyright..........2
2. Introduction..........3
2.1 New Versions of this Document..........3
2.2 Feedback and Corrections..........3
2.3 Acknowledgements..........3
3. Requirements..........3
3.1 Which Hardware is supported..........3
3.2 Which Software is needed..........4
3.3 Which Software is supported..........4
4. Tablets as Mouse−Replacement for the Linux−Console..........5
4.1 Tablets and gpm..........5
5. Tablets with XFree 3 and XFree 4..........6
5.1 Required XFree Version..........6
5.2 Sample XF86Config snippet for XFree 3..........6
5.3 Sample XF86Config snippet for XFree 4..........7
5.4 Syntax Difference between XFree 3 and XFree 4..........8
5.5 Section Module..........8
5.6 Device Type (XF3) and Type (XF4)..........8
5.7 DeviceName (XF3) and Identifier (XF4)..........9
5.8 Port (XF3) and Device (XF4)..........9
5.9 DebugLevel..........9
5.10 Serial Num (intuos series only)..........9
How to find out the serial number of a device..........10
Serial Num − Example of what has been logged (XFree 3)..........10
5.11 Mode absolute, Mode relative..........11
5.12 Device Modes Extension, Core, AlwaysCore..........11
5.13 HistorySize..........12
5.14 Suppress..........12
5.15 TiltMode (XF3) and Tilt (XF4)..........13
5.16 MaxX, MaxY, MaxZ (since alpha 20)..........13
5.17 ResolutionX, ResolutionY, ResolutionZ (since alpha 20)..........13
5.18 TopX, TopY, BottomX, BottomY Statements..........14
Getting the maximal X, Y Values, current configuration, and the resolution..........14
Getting Resolution − Example of what has been logged..........14
5.19 KeepShape..........15
5.20 Threshold..........15

Table of Contents

5.21 USB (since alpha 20)....16
6. Changing configuration from within a running X−Server....16
6.1 Setting up the Gimp to use XInput devices....16
6.2 The Switch Device....17
6.3 Buttons....17
6.4 Button Mapping with xmodmap for Devices in Core Mode....17
6.5 Button Mapping with xinput for Devices in Extension Mode....17
6.6 xsetmode − changing absolute / relative mode....18
6.7 xsetpointer − setting the default core device....18
6.8 Keys....18
6.9 Utilities to use more than one tablet and for toggle support on / off....18
7. XFree FAQ − Problems And Questions....19
7.1 Gnome and Gimp only work like a usual mouse, without pressure and tilt. Gimp Input Devices Dialog shows "....19
7.2 Is the wheel of the mouse (cursor) supported ?....19
7.3 How can I find out the intuos Serial Num of my tool ?....19
7.4 How do I change the action of each button of my tool?....19
7.5 How can I use the intuos/artpad/ultrapad macro area ?....19
7.6 Is it possible to define multiple active areas on the same tablet ?....20
7.7 Can I use more than one tablet at the same time ?....20
7.8 The Wacom Driver eats my CPU Time. Any Hint ?....20
8. Gpm FAQ − Problems And Questions....20
8.1 NCurses programs only support cut and paste with my new gpm....20
9. Further Information / Used Documents....20

Wacom Tablet HOWTO

Stefan Runkel <runkel@runkeledv.de>

Version 1.0.7 of Dec 29, 2000

Installation of (not only) Wacom graphic tablets under linux and / or xfree86

---

1. Copyright

2. Introduction

- 2.1 New Versions of this Document
- 2.2 Feedback and Corrections
- 2.3 Acknowledgements

3. Requirements

- 3.1 Which Hardware is supported
- 3.2 Which Software is needed
- 3.3 Which Software is supported

4. Tablets as Mouse−Replacement for the Linux−Console

- 4.1 Tablets and gpm

5. Tablets with XFree 3 and XFree 4

- 5.1 Required XFree Version
- 5.2 Sample XF86Config snippet for XFree 3
- 5.3 Sample XF86Config snippet for XFree 4
- 5.4 Syntax Difference between XFree 3 and XFree 4
- 5.5 Section Module
- 5.6 Device Type (XF3) and Type (XF4)
- 5.7 DeviceName (XF3) and Identifier (XF4)
- 5.8 Port (XF3) and Device (XF4)
- 5.9 DebugLevel
- 5.10 Serial Num (intuos series only)
- 5.11 Mode absolute, Mode relative
- 5.12 Device Modes Extension, Core, AlwaysCore
- 5.13 HistorySize
- 5.14 Suppress
- 5.15 TiltMode (XF3) and Tilt (XF4)
- 5.16 MaxX, MaxY, MaxZ (since alpha 20)

- 5.17 ResolutionX, ResolutionY, ResolutionZ (since alpha 20)
- 5.18 TopX, TopY, BottomX, BottomY Statements
- 5.19 KeepShape
- 5.20 Threshold
- 5.21 USB (since alpha 20)

6. Changing configuration from within a running X−Server

- 6.1 Setting up the Gimp to use XInput devices
- 6.2 The Switch Device
- 6.3 Buttons
- 6.4 Button Mapping with xmodmap for Devices in Core Mode
- 6.5 Button Mapping with xinput for Devices in Extension Mode
- 6.6 xsetmode − changing absolute / relative mode
- 6.7 xsetpointer − setting the default core device
- 6.8 Keys
- 6.9 Utilities to use more than one tablet and for toggle support on / off

7. XFree FAQ − Problems And Questions

- 7.1 Gnome and Gimp only work like a usual mouse, without pressure and tilt.
- 7.2 Is the wheel of the mouse (cursor) supported ?
- 7.3 How can I find out the intuos Serial Num of my tool ?
- 7.4 How do I change the action of each button of my tool?
- 7.5 How can I use the intuos/artpad/ultrapad macro area ?
- 7.6 Is it possible to define multiple active areas on the same tablet ?
- 7.7 Can I use more than one tablet at the same time ?
- 7.8 The Wacom Driver eats my CPU Time. Any Hint ?

8. Gpm FAQ − Problems And Questions

- 8.1 NCurses programs only support cut and paste with my new gpm

9. Further Information / Used Documents

---

1. Copyright

---

2. Introduction

Graphic tablets are now quite inexpensive and can help allot with the work of image manipulation. In addition to that what a mouse can do for you they have the following features:

- pressure and tilt sensitivity
- high resolution motion control
- sub−pixel positioning
- an input device that looks like and is handled like a normal Pen.

This document addresses mainly the products from Wacom, because I own 3 different products of them and because their products are highly available in the region of Europe.

Nevertheless I think it should be easy to adapt the shown concepts to other products.

I have written this document because I have not found a document about this subject which contained completely what I wanted to know. So I got small parts of information from various people in the net and assembled them to this HOWTO. Regarding the questions of my readers, I must realize that this document also is not complete and never will be. But from the same questions I can tell that it helped many people on the way, and each new question lets this HOWTO grow.

2.1 New Versions of this Document

The latest version of this document will be available at http://www.runkeledv.de/linux.htm .

2.2 Feedback and Corrections

Comments, corrections, and additions are greatly be appreciated. I can be contacted at:

Runkel@runkeledv.de or at RunkelEDV@t−online.de

2.3 Acknowledgements

I would like to thank all authors listed in the Further Information / used documents Section as well as everyone who answered my questions by email. This HOWTO reflects much of their work.

---

3. Requirements

This chapter is about what is needed to get your tablet working.

3.1 Which Hardware is supported

First of all, you should have a tablet, of course. The following Wacom tablets are supported:

- ArtZ II series (known in Europe as the UltraPad series − the same UltraPad name also was used for earlier tablets that only work partially.)

- ArtPad II
- PL300 (combined LCD screen and tablet)
- Pen Partner
- Graphire alias FAVO (mouse and pen, get at least the alpha 16 driver for full support from http://www.lepied.com/xfree86/).
- Intuos series (support may be only partial with the driver of your distribution, get updated alpha versions from Frederic Lepied's page: http://www.lepied.com/xfree86/)

The older SD and HD series are not supported by the standard XFree86 driver, however, a modified driver that supports these devices as well some OEM products with embedded screens including the IBM Thinkpad 360 PE and Toshiba T200 is available from:
http://hwr.nici.kun.nl/pen−computing/pen−computing−formats.html

3.2 Which Software is needed

- If you want support for the linux console, get the latest version of gpm from ftp://ftp.prosa.it/pub/gpm or from ftp://animal.unipv.it/pub/gpm (mirror)

- If you want support for XFree86, use at least version 3.3.3.1 or get the latest from http://www.xfree.org

3.3 Which Software is supported

- For the linux console, the only program I know is gpm.

- For XFree, the keyword is XInput. This specification has to be supported by device drivers which provide extra information. In turn, XInput has to be understood by programs which want to use alternative pointer devices.

There is a big number of programs based on the gtk library. The gtk has XInput support and makes it very easy to use.
At least the following applications support XInput:

- The Gimp − a powerful image manipulation program in the style of Photoshop. The 1.1.x instable development version includes XInput support as a standard feature. We are awaiting the next stable release, Version 1.2.x. Obtain Gimp from http://www.gimp.org , it really deserves it ! I am sure most of you know this and it was the reason to buy a tablet.

- gsumi − a simple B/W drawing program that supports drawing / erasing with pressure and tilt sensitivity. Get it from the gsumi web page: http://www.gtk.org/~otaylor/gsumi/

- xink (By Ralph Levien) − another rudimentary drawing program for X.

  xink is available from: ftp://kiwi.cs.berkeley.edu/pub/raph/xink.tar.gz

- RasMol − a molecular visualization program that supports a hardware dial box using XInput. look at http://www.umass.edu/microbio/rasmol/distrib/rasman.htm .

- xinput (by Frederic Lepied) − a very useful utility for configuring and testing XInput devices. xinput is available from: ftp://ftp.x.org/contrib/utilities/xinput−1.2.tar.gz

---

4. Tables as Mouse−Replacement for the Linux−Console

4.1 Tablets and gpm

The gpm daemon supports allot of input devices, including tablets. If you want a list of the supported devices of your gpm, do a

```
gpm −t help
```

and you get it.

The Wacom driver of gpm 1.18.0 and below seems to be for the very old artpad models, the documentation does not say anything about this. I tested it with ultrapad, penpartner, graphire and intuos models but it did not work. I have rewritten the wacom driver for gpm, to use it you need at least gpm version 1.18.2 . Since 1.18.2, the old wacom driver has been replaced with my driver for wacom graphire, penpartner and ultrapad products.

However, if you own an old artpad, use gpm < 1.18.1 and try:

```
killall gpm ; gpm −t wacom −m /dev/ttyS0
```

and it should work.

If you own a graphire, use gpm > 1.18.1 instead and use

for the relative (mouse−like) mode:

```
killall gpm ; gpm −t wacom −o relative −m /dev/ttyS0
```

for the absolute (tablet−like) mode:

```
killall gpm ; gpm −t wacom −o absolute −m /dev/ttyS0
```

Of course, you get only the buttons and the movement function, pressure and tilt are not supported because they are not too useful in text−based applications.
There is no support for ultrapad macro buttons or graphire mouse wheel.

If you use gpm and XFree86, you may or may not run in problems, because normally a device can be opened only by one program at a time. There are two solutions for that:

- The gpm repeater mode. If you start gpm with the "−R protocol" option, it will repeat all data it gets to the device /dev/gpmdata. Let us say, you have a ps2 type mouse and your gpm call looks like: *gpm −t ps2 −Rmman* then gpm gets your ps2 mouse data and tries to translate this into a mman packet, which it will emit to /dev/gpmdata. Unfortunately, gpm can translate to only a few protocols at the time of writing. Since gpm 1.18.1, these protocols to which can be translated are marked with an asterix at the first column. Again, do a *gpm −t help* to get the list.

If you want to omit the translation and only get the data byte for byte, just call *gpm −t ps2 −R raw* . You can then make XFree use that second device by putting the line *Device "/dev/gpmdata"* in your XF86Config in the pointer section or the XInput section.

- If you use startx, you can modify it that it does a *gpm −k* before the X−Server is started. This option kills a running gpm.

---

5. Tablets with XFree 3 and XFree 4

5.1 Required XFree Version

Minimum XFree version should be 3.3.3.1 because this version has extended XInput support, the second side switch works now and the blocking mouse effect has been fixed.

If you must use any prior version, XInput support should be available since 3.1.2.d.

XInput support in XFree is configured basically in a file called XF86Config. In most cases the XF86Config file is located in /etc or in /etc/x11. I will show now two very short configuration sections, the first works with XFree 3, the other is for XFree 4. I will explain then what can be changed and why this is needed.

5.2 Sample XF86Config snippet for XFree 3

```
Section "Module"
# Wacom Device Driver
Load "xf86Wacom.so"
EndSection

Section "XInput"
# Settings for Wacom pen
  SubSection "WacomStylus"    # type of input device
    Port "/dev/ttyS0"         # Serial Port
    DeviceName "Pen1"         # Name, choose it free
    Baudrate 9600             # Usable only for intuos
    Mode Absolute             # Map Tablet Area to Screen
    AlwaysCore                # See text
    HistorySize 200           # Buffer size for motion events
  EndSubSection

# Settings for Wacom eraser
  SubSection "WacomEraser"    # type of input device
    Port "/dev/ttyS0"         # Serial Port
    DeviceName "Eraser1"      # Name, choose it free
    Baudrate 9600             # Usable only for intuos
    Mode Absolute             # Map Tablet Area to Screen
    AlwaysCore                # See text
    HistorySize 200           # Buffer size for motion events
  EndSubSection

# Settings for Wacom cursor (mouse)
```

```
  SubSection "WacomCursor"    # type of input device
    Port "/dev/ttyS0"         # Serial Port
    DeviceName "Cursor"       # Name, choose it free
    Baudrate 9600             # Usable only for intuos
    Mode Absolute             # Map Tablet Area to Screen
    AlwaysCore                # See text
    HistorySize 200           # Buffer size for motion events
  EndSubSection

EndSection
```

5.3 Sample XF86Config snippet for XFree 4

```
# declare input devices, later defined
Section "ServerLayout"
        Identifier     "XFree86 Configured"
        Screen      0  "Screen0" 0 0
        InputDevice    "Mouse0"    "CorePointer"
        InputDevice    "Keyboard0" "CoreKeyboard"
        InputDevice    "pen1"      "AlwaysCore"
        InputDevice    "eraser1"   "AlwaysCore"
        InputDevice    "cursor1"   "AlwaysCore"
EndSection

Section "Module"
        Load  "extmod"
        Load  "xie"
        Load  "pex5"
        Load  "glx"
        Load  "dri"
        Load  "GLcore"
        Load  "dbe"
        Load  "record"
        Load  "wacom"       # Wacom Device Driver
EndSection

# Settings for wacom pen
Section "InputDevice"
        Identifier   "pen1"
        Driver       "wacom"
        Option       "Type" "stylus"
        Option       "Device" "/dev/ttyS0"
Endsection
# Settings for wacom eraser
Section "InputDevice"
        Identifier   "eraser1"
        Driver       "wacom"
        Option       "Type" "eraser"
        Option       "Device" "/dev/ttyS0"
Endsection
# Settings for wacom cursor (mouse)
Section "InputDevice"
        Identifier   "cursor1"
        Driver       "wacom"
        Option       "Type" "cursor"
        Option       "Device" "/dev/ttyS0"
Endsection
```

5.4 Syntax Difference between XFree 3 and XFree 4

Please notice the different syntax between XFree 3 and XFree 4:
In XFree 3 configuration is done in the section "XInput" with one subsection for each device. XFree 3 has keywords.
In XFree 4 configuration is done with one section "InputDevice" for each device. All Settings are "options", with one exception: the identifier statement. The Alwayscore option goes to the Serverlayout section.

Because the usage can be seen from the above examples, this difference will be considered as clear for the rest of the document.

5.5 Section Module

XInput support in XFree is reached due to the use of modules. The wacom device driver module file for XFree 3 is called xf86Wacom.so, for XFree 4 the filename changed to wacom_drv.o. The files are located usually in /usr/X11R6/lib/modules/ or in /usr/X11R6/lib/modules/input. We have to specify in the "Module" section that we want to load this driver (look at above snippets):

XFree 3: Load "xf86Wacom.so"
XFree 4: Load "wacom"

This has to appear only once per XF86Config file.

5.6 Device Type (XF3) and Type (XF4)

A physical device can be:

- The tip of your pen
- The eraser of your pen
- A mouse like the Wacom products lens cursor, 4d−mouse, graphire−mouse
- The airbrush

The penpartner and graphire products simply recognize the type of a device, but they can not distinguish them. For example, you can not use two different pens, say, one configured red and one configured blue − all what these products say is: "I found a pen tip on my surface", or "I found an Eraser on my surface", or "I found a mouse on my surface".

XInput maps the device types to devices, these device types are later specified in XF86Config. For XFree 3 this is a subsection in the section XInput, for XFree 4 this goes to individuell InputDevice sections.
The devicetypes are:

- `WacomStylus` Tip of a pen or airbrush
- `WacomEraser` Eraser of a pen or airbrush
- `WacomCursor` Mice, like graphire mouse, lens cursor, 4d−mouse

As you can see from above examples, device types are specified with the keyword "Devicetype" in XFree 3 and with Option "Type" in XFree 4.

5.7 DeviceName (XF3) and Identifier (XF4)

You must give every configured device a unique name. This name will be used later in your program to talk to that device. You can choose this name free. This statement is mandantory. In XFree 3 you will use the "Devicename" keyword, in XFree 4 you will use the Identifier "name" keyword. Please notice that Identifier is not an Option !

5.8 Port (XF3) and Device (XF4)

XInput wants to know at which serial port this device can be found. This allows me to use two different tablets at the same time, for example a graphire connected to /dev/ttyS0 and a penpartner connected to /dev/ttyS1. USB is treated as a serial line too, so this statement is mandantory. If you connect your product to the USB, you have to specify this with Option USB.

5.9 DebugLevel

This statement controls how verbose the Wacom driver is. The Level goes from 0 to 10. If this statement is not given, DebugLevel 0 is used, which logs the very less.

Example (XFree 3):

```
SubSection "WacomStylus" # Pen
  DeviceName "PenRed"    # Name, choose it free
  ...
  DebugLevel 10          # be very verbose
  ...
EndSubSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"            # Name, choose it free
  Option     "Type" "stylus"
  ...
  Option "DebugLevel" "10"       # be very verbose
  ...
EndSection
```

5.10 Serial Num (intuos series only)

The intuos product is more efficient than penpartner and graphire because it can distinguish many devices of the same type. This means that it is now possible to use two pens, one configured red and one configured blue. As you see, the use of device types is not enough information to handle different pens. Therefore, each intuos device, be it a pen, ink pen, mouse, or whatever has a serial number, which you can specify to aid XInput in resolving the right device. This would look like:

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus" # Pen
```

```
    DeviceName "PenRed"    # Name, choose it free
    ...
    Serial 2609917443      # Serial Number of that device
    ...
  EndSubSection
  SubSection "WacomStylus" # another Pen
    DeviceName "PenBlue"   # Name, choose it free
    ...
    Serial 2609918664      # Serial Number of that device
    ...
  EndSubSection
EndSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"    # Name, choose it free
  Option     "Type"      "stylus"
  ...
  Option     "Serial"    "2609917443"
  ...
EndSection
Section "InputDevice"
  Identifier "PenBlue"   # Name, choose it free
  Option     "Type"      "stylus"
  ...
  Option     "Serial"    "2609918664"
  ...
EndSection
```

It should be said that combined input devices like pens have only one serial number. The driver uses the serial number it gets to recognize one specific pen and the device type to distinguish the tip of that specific pen from its eraser.

How to find out the serial number of a device

1. Set DebugLevel to 6
2. Start the X−Server by typing *X 2>t*
3. Tip all devices down on the tablet and make a notice of the order you tipped them.
4. Kill the X−Server (usually Ctrl+Alt+Backspace)
5. Do a *grep serial_num t*

You should get a list of your different numbers.

Serial Num − Example of what has been logged (XFree 3)

```
BEGIN xf86WcmProc dev=0x8354d60 priv=0x833e3f0 type=stylus flags=9 what=1
xf86WcmProc pWcm=0x8354d60 what=ON
END xf86WcmProc Success what=1 dev=0x8354d60 priv=0x833e3f0
device_id=0x96 serial_num=2595227137 type=cursor
[cursor] abs prox=false x=0 y=0 z=0 button=false buttons=0
```

(serial_num is in Line 4 Word 2).

5.11 Mode absolute, Mode relative

If you set a device in mode absolute, this means, that the active area of the tablet will be mapped to the screen. Every time you go down to the tablet at the same point with an absolute device the pointer will appear at the same point of the screen.

If you set a device in mode relative, you will get the well known behavior of a mouse. This means, that if you take the mouse off from the surface, move it and go down again, the pointer does (ideally) not move.

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus"   # Pen
    DeviceName "PenRed"      # Name, choose it free
    ...
    Mode absolute
    ...
  EndSubSection
EndSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"        # Name, choose it free
  Option     "Type"          "stylus"
  ...
  Option     "Mode"          "absolute"
  ...
EndSection
```

5.12 Device Modes Extension, Core, AlwaysCore

XFree knows two pointers: one with only the standard features (buttons, moving−capabilities), which is used for selecting menus, text, clicking buttons and doing other controlling stuff. This is the *Core Device*.

The other pointer is used by applications which want more information, like pressure and tilt.
This is the extension device.

Starting with version 3.3.3.1, the Statement AlwaysCore tells the driver that it should send both types of events.

If you do not specify Alwayscore in your XF86Config, then this device is initially used as extension device, this means it is usable only in applications but you can not control the menus of your window manager with it.

If Alwayscore is given, then your device acts as core pointer (in addition to the mouse) as well as it sends tilt and pressure information to applications which opened the device in extension mode.

You can configure two different logical devices, one in Core−Mode and one in Extension−Mode, to the same physical device.
To do so, simply type two identical sections, only the DeviceName statement must differ. Then specify AlwaysCore only for the last section.

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus"   # Pen
    DeviceName "PenRed"      # Name, choose it free
    ...
    AlwaysCore
    ...
  EndSubSection
EndSection
```

For XFree 4, this is not an option in the InputDevice section, it goes to the Serverlayout section and can be one of: AlwaysCore / SendCoreEvents or CorePointer:

Example (XFree 4):

```
Section "ServerLayout"
        Identifier     "XFree86 Configured"
        Screen      0  "Screen0" 0 0
        ...
        InputDevice    "icursor"   "AlwaysCore"
EndSection
```

5.13 HistorySize

This statement sets the buffer size that is used to cache motion events.

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus"   # Pen
    DeviceName "PenRed"      # Name, choose it free
    ...
    HistorySize 200
    ...
  EndSubSection
EndSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"      # Name, choose it free
  Option     "Type"        "stylus"
  ...
  Option     "HistorySize" "200"
  ...
EndSection
```

5.14 Suppress

This statement specifies how many units the device must move before the driver moves the pointer. This can be necessary if very large resolutions are used.

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus"    # Pen
    DeviceName "PenRed"       # Name, choose it free
    ...
    Suppress 10
    ...
  EndSubSection
EndSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"    # Name, choose it free
  Option     "Type"      "stylus"
  ...
  Option     "Suppress"  "10"
  ...
EndSection
```

5.15 TiltMode (XF3) and Tilt (XF4)

This works only with intuos series tablets.
TiltMode enables sending of tilt information for intuos devices. If this statement is missing, only pressure information is transferred.

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus"    # Pen
    DeviceName "PenRed"       # Name, choose it free
    ...
    TiltMode
    ...
  EndSubSection
EndSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"    # Name, choose it free
  Option     "Type"      "stylus"
  ...
  Option     "Tilt"      "1"
  ...
EndSection
```

5.16 MaxX, MaxY, MaxZ (since alpha 20)

Max[axis] limits / owerwrites the max[axis] values normaly retrieved from the tablet. More or less developer stuff.

5.17 ResolutionX, ResolutionY, ResolutionZ (since alpha 20)

Resolution[axis] presets the resolution[axis] values normaly retrieved from the tablet. More or less developer stuff.

5.18 TopX, TopY, BottomX, BottomY Statements

These four statements allow to reduce the active area of the tablet. My intuos A4 oversize, for example, is so big that you can not do painting work with it because the ways of the pen are too long. I use these four Statements to tell the driver that it should map only the lower left quarter of the tablet to the screen.
It is possible to define more than one active area on the same tablet: Define as many subsections as you need with the same Port Statement but with different Top... statements and Devicenames. This results in multiple logical devices all of them restricted to their individual bounds. You may use one device in absolute mode and one in relative mode as well.

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus"   # Pen
    DeviceName "PenRed"      # Name, choose it free
    ...
    TopX 0
    TopY 5000
    BottomX 5000
    BottomY 10000
    ...
  EndSubSection
EndSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"    # Name, choose it free
  Option     "Type"      "stylus"
  ...
  Option     "TopX"      "0"
  Option     "TopY"      "5000"
  Option     "BottomX"   "5000"
  Option     "BottomY"   "10000"
  ...
EndSection
```

Getting the maximal X, Y Values, current configuration, and the resolution

If you use the above four statements, you sometimes want to make some calculations on the size and position of the rectangle to be defined. Often the maximal tablet coordinates and the resolution are needed for this. To get these values, start your server: *X 2>t*. Kill the X−Server (usually Ctrl+Alt+Backspace) and do a *grep "X=" t* .You should get back the values in question.

Getting Resolution − Example of what has been logged

```
(−−) Wacom IV tablet maximum X=5103 maximum Y=3711 X
     resolution=1000 Y resolution=1000 suppress=6
(−−) Wacom tablet top X=0 top Y=0 bottom X=5103 bottom Y=3711
(−−) Wacom tablet top X=0 top Y=0 bottom X=5103 bottom Y=3711
(−−) Wacom tablet top X=0 top Y=0 bottom X=5103 bottom Y=3711
```

5.19 KeepShape

This option uses the TopX, TopY statements (or their built in defaults, if omitted) and adjusts the BottomX, BottomY statements so that

- The ratio height / width of the screen is the same ratio on the tablet
- The active area starting at TopX, TopY of the tablet is as big as it can be with the above condition.

Any given BottomX, BottomY statement is ignored, because these values will be calculated.

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus"   # Pen
    DeviceName "PenRed"      # Name, choose it free
    ...
    KeepShape
    ...
  EndSubSection
EndSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"    # Name, choose it free
  Option     "Type"      "stylus"
  ...
  Option     "KeepShape" "1"
  ...
EndSection
```

5.20 Threshold

This is available since alpha 16 and is used to control the minimum pressure that is needed to detect that a tool is pressed on the surface. It works with intuos and graphire. Please note that the integer value has to be coded as String, unlike all other integer values.

Example (XFree 3):

```
Section "XInput"
  SubSection "WacomStylus"   # Pen
    DeviceName "PenRed"      # Name, choose it free
    ...
    Threshold "5"
    ...
  EndSubSection
EndSection
```

Example (XFree 4):

```
Section "InputDevice"
  Identifier "PenRed"   # Name, choose it free
  Option     "Type"     "stylus"
  ...
  Option     "Treshold" "5"
```

```
  ...
EndSection
```

5.21 USB (since alpha 20)

USB tells the driver that the serial line is the usb port. This needs a Kernel 2.4 with usb enabled. Alpha 20 driver only supports absolute devices. Please read the Wacom−USB−mini−HOWTO by Arnauld Claden for more information on USB setup.

---

6. Changing configuration from within a running X−Server

6.1 Setting up the Gimp to use XInput devices

Gimp has built−in XInput support since Version 1.1.x.

Gimp must know which devices it should use and in what mode. You have to open the *File/Dialogs/Input Devices* − dialog for setting this up. You will find two listbox−controls at the top of that window labeled *Device* and *Mode* . Choose the device to set up from the *Device* control and choose a mode from the *Mode* −control.

The modes are:

- disabled: Gimp does not use this device.
- window:
  Gimp uses this device, the cursor is drawn by the application. The cursor can not leave the active window. The active area of the tablet is mapped to this window.
- screen:
  Gimp uses this device. The cursor is drawn by the X−Server. The cursor can move over the whole screen. The active area of the tablet is mapped to the screen.

If you see two cursors linked to your device, you have two solutions:

- Use mode screen but you may or may not loose the special gimp−tool cursors; If so, all cursors will be replaced by the pen cursor.
- Configure two different devices in the XF86Config file, one *AlwaysCore* and one not. Tell gimp to use the later one in mode window. If you do so and have one of the statements *TopX, TopY, BottomX, BottomY* in your XF86Config, take care to use the same adjustments for that statements in both logical devices. If you do not, then you get confused because X draws a cursor of the one device while the other is active. the effect of this is a cursor, very far away from the point of action (hotspot).

Below the two listbox−controls there is a tab−control with the two register tabs *Axes* and *Keys* . *Axes* assigns an axis a function, it is mostly not necessary to change this. But think about a touch−screen device which is built in a table rotated by 90 degrees, then you might want to swap the x and y axis.

Some tablets have so called macro keys at the top which may hold some often used functions. The *Keys* tab lets you assign a character to a macro key. For example, you can put Ctrl+Shift+R, to a macro key. If you activate this key, then the rulers are toggled.

The current Wacom driver supports only the macro keys of the ultrapad series, the macro area of the intuos products is not yet usable for that.

Now we should talk about how the devices can be used. Open the *File/Dialogs/Device Status* −dialog. Open an image.

You set up each device independently from each other in gimp.

If you move the cursor with different devices in the image window you can see the devices change in the Device Status dialog. If you pick a tool, brush, pattern or color with a device, again the changes are reflected in the dialog. You can save the settings in the Status Dialog, so that they will be restored before your next session.

6.2 The Switch Device

This is a special device that is always present. It generates an event every time a new device becomes the core pointer. It has a pseudo "axis". The "value" of this axis is the id of the core pointer device. I do not know what a user can do with it − this is mainly helpful for internal use.

6.3 Buttons

The buttons of the devices are as different as the devices are:
A pen has at least a tip, but it may have one or two side switches and an eraser. A mouse may have up to 32 buttons (but usually 3). Buttons are numbered from 1 to the number of buttons. With the next two utilities you can change, to what number a button is mapped.

6.4 Button Mapping with xmodmap for Devices in Core Mode

xmodmap will only modify the Core Pointer. As there is only one Core pointer at a time, it makes no difference between the physical devices which may become the Core Pointer. For the moment, let us forget the whole XInput stuff and think of a left−hander who just wants to swap the left and right mouse buttons. You would execute *xmodmap −pp* to look what the current assignment is. You should get the following table back:

```
Physical         Button
Button           Code
1               1      # ( Left   Button )
2               2      # ( Right  Button )
3               3      # ( Middle Button )
```

To swap the buttons, you do a *xmodmap −e "pointer = 2 1 3"* , and to get back, *xmodmap −e "pointer = default"* . This should work with every device with at least two buttons. Note that the term *"pointer = x x x"* has to be quoted to prevent it from being changed by the shell.

6.5 Button Mapping with xinput for Devices in Extension Mode

Back to XInput now. If you use the gimp, you may want to change the button mapping for each device separately (may be you are happy with the mouse but want to swap the two side−switches of the pen). Frederic Lepied has written a utility called xinput for that.

To swap the side−switches, you would do a *xinput list* to get a list of the devices and their current settings. Swapping is done with *xinput set−button−map Pen1 1 3 2* where *Pen1* is the Device to change.

Starting with XFree 3.3.2 this works also with devices that are configured *AlwaysCore* in XF86Config.

6.6 xsetmode − changing absolute / relative mode

With xsetmode you can change the mode of a device between absolute and relative.

Example:
xsetmode GraphireMouse ABSOLUTE .

6.7 xsetpointer − setting the default core device

If you have none of your devices configured *AlwaysCore* and you want a device to become the core−pointer, then xsetpointer must be used.

Do a *xsetpointer Devicename* . The old core−device (usually the mouse) is not usable anymore and the one you specified should be active. For example I can make the graphire mouse the standard core device from within a running XFree.

xsetpointer −l lists all devices and the modes they are in.

6.8 Keys

Some devices have macro keys or pads on them, to which a scancode or string may be assigned. This works only for devices in extension mode and is therefore done in the application you want use the keys with.

xinput −l gives information about the number of keys and things like the first scancode.

6.9 Utilities to use more than one tablet and for toggle support on / off

I own myself 3 tablets of Wacom and use them on my notebook. Whenever no tablets are connected and I start X, I have to wait very long until the driver gives up. If xdm is used, this increases to multiple of that timeouts.

I have written two utilities which should make live easier with that:

- The shell script **sx** for those who use startx. sx uses the dialog tool to present a nice menu where the user can choose:

- ♦ one of up to 10 devices (tablet, joystick, ...)
  - ♦ for notebooks, which display to use (internal or external display)
  - ♦ which window−manager to use
- The small gtk−application **xinput−chooser** for those who use xdm. xinput−chooser presents the user a menu whenever the xdm login screen is shown, from which one of up to ten different configurations can be chosen.

The concept behind this is that a XF86Config file is split in at least a XF86Config.bare file, which holds the information common to all configurations, and various snippets, each of them holding the special information for a particular configuration.

The two utilities will then concat the parts back to a working configuration.

You can get this packet from http://www.runkeledv.de/download , it is named xinput_chooser_sr.

---

7. XFree FAQ − Problems And Questions

7.1 Gnome and Gimp only work like a usual mouse, without pressure and tilt. Gimp Input Devices Dialog shows "No Input Devices"

It seems that you have not compiled your gtk with XInput−support. Please rebuild gtk and use *./configure −−with−xinput=xfree* for building the Makefile. Then do a *make && make install*. Then rebuild your gimp so that it uses this extendet gtk.

7.2 Is the wheel of the mouse (cursor) supported ?

Yes, it generates button 4 and button 5 events.

7.3 How can I find out the intuos Serial Num of my tool ?

read the section about Serial_Num

7.4 How do I change the action of each button of my tool?

Read the section about buttons.

7.5 How can I use the intuos/artpad/ultrapad macro area ?

You can not use the intuos macro area, sorry.
Only Artpad / Ultrapad Macro stripes work. Read the section about Gimp Keys.

7.6 Is it possible to define multiple active areas on the same tablet ?

Yes, read the section about the Top... Statements

7.7 Can I use more than one tablet at the same time ?

Yes, read the section about the Port Statement

7.8 The Wacom Driver eats my CPU Time. Any Hint ?

I was told, that hdparm −u1 /dev/hdx (replace x with your drive letter) may help.

---

8. Gpm FAQ − Problems And Questions

8.1 NCurses programs only support cut and paste with my new gpm

You probably mixed a new gpm with an old gpm library. Or You use an old client which searches the socket gpmctl in /var/run instead in /dev. In the later case you have two options: get new clients and recompile them or change gpm.h so that it uses /var/run instead of /dev and recompile gpm and libgpm.

---

9. Further Information / Used Documents

gpm:
gpm man page, gpm FAQ, source code file mice.c

XInput:
XInput HOWTO by Owen Taylor

XFree and Wacom:
XF86Config man page

Frederic Lepied's site

Wacom−USB−mini−HOWTO by Arnauld Claden much, much, e−mails...
