The Webcam HOWTO

Howard Shane

<hshane[AT]austin.rr.com>

Revision History

Revision 1.61	2005−02−21	Revised by: jhs
Update on revived Philips Webcam driver development		
Revision 1.6	2005−01−02	Revised by: jhs
Errata fixed, some rewrites for readability, new chipsets and updates		
Revision 1.1	2004−01−12	Revised by: jhs
Update for 2.6 series kernel release and info on NW802−based webcams		
Revision 1.0	2003−12−04	Revised by: JP
Initial Release / Reviewed by TLDP		
Revision 0.5	2003−11−07	Revised by: jhs
Final revision after v4l mailing list feedback		
Revision 0.1	2003−10−12	Revised by: jhs
Initial draft posted		

This document was written to assist the reader in the steps necessary to configure and use a webcam within the Linux operating system.

Table of Contents

1. Introduction.....1
- 1.1. Copyright Information.....1
- 1.2. Disclaimer.....1
- 1.3. New Versions.....1
- 1.4. Credits.....1
- 1.5. Feedback.....2
- 1.6. Conventions Used in this Document.....2

2. Enabling Support for Your (Webcam) Hardware in Linux.....3
- 2.1. Drivers and Modules.....3
  - 2.1.1. Module or In−Kernel?.....3
  - 2.1.2. Patching, Source−Only or Precompiled Binary?.....4
- 2.2. Supporting the Connection Type.....4
  - 2.2.1. USB Webcams.....4
  - 2.2.2. IEEE 1394 (Firewire", i.Link").....6
  - 2.2.3. Generic Parallel Port Support for Parport Webcams.....6
- 2.3. Specific Webcam Models.....6
  - 2.3.1. 3com HomeConnect PC Digital Webcam.....7
  - 2.3.2. CPiA based Webcams.....7
  - 2.3.3. SE401, SE402 and EP800 based USB webcams.....7
  - 2.3.4. OmniVision based Webcams.....8
  - 2.3.5. Logitech (formerly Connectix) Quickcam Support.....9
  - 2.3.6. ICM532 Based Webcams.....9
  - 2.3.7. NW802 Based Webcams.....10
  - 2.3.8. Philips USB Webcams.....10
  - 2.3.9. SPCA50X USB Camera Linux Driver.....10
  - 2.3.10. STV0680 based Models.....11
  - 2.3.11. Winbond w9966cf.....11
  - 2.3.12. Xirlink C−it" HDCS−1000 based Webcams.....11

3. Accessing the Video Device.....12
- 3.1. The Video Devices Node.....12
- 3.2. Groups and Permissions.....13

4. Framegrabbing Applications.....14
- 4.1. Command Line Programs.....14
  - 4.1.1. Streamer.....14
  - 4.1.2. camE.....14
  - 4.1.3. Motion.....14
  - 4.1.4. Webcam.....15
  - 4.1.5. SANE.....15
- 4.2. GUI(Grapical User Interface)−Based Programs.....15
  - 4.2.1. Xawtv.....15
  - 4.2.2. Gqcam.....15
  - 4.2.3. Camorama.....15
  - 4.2.4. GnomeMeeting.....16

Table of Contents

5. Troubleshooting....17
- 5.1. Help, I have a USB webcam and don't know exactly what model it is and/or who the manufacturer is. What do I do?....17
- 5.2. Help, I can't find the camera device in /dev!....17
- 5.3. Help, I can see the camera device (both in person and as a device node in /dev), but I can't access it!....17
- 5.4. Help, my camera has a driver that is source−only, i.e., has to be built by me! Where do I start?....18
- 5.5. I am using Debian GNU/Linux. Is there an easier way to go through all this kernel compiling stuff and building of source modules?....18
- 5.6. Help, my camera is supported by a driver that has to be patched into my kernel! What do I do?....19
- 5.7. Help−−as in, where can I get more of it?....19
- 5.8. Help, I want to contribute to Video4Linux support in Linux! Who do I get in touch with?....19

A. Gnu Free Documentation License....20

1. Introduction

This document was written to assist the reader in setting up and configuring a webcam, digital camera, or other video device in the Linux operating system. It outlines how to enable the necessary kernel and/or software support and various frame−grabber applications that can be used to access your device. It does not discuss the differences in graphic and video formats, the features and/or capabilities of particular devices, or the encoding or conversion of video formats.

1.1. Copyright Information

1.2. Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and other content entirely at your own risk. As this is a new edition, there may be technical or other inaccuracies that may result in the loss of irreplaceable data. In any case, proceed with caution, and realize that although errors are highly unlikely, the author can accept no responsibility for them.

All copyrights are held by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

1.3. New Versions

This is the fourth release.

The latest version number of this document can be found here.

1.4. Credits

I would like to thank all of the individuals that have pioneered video support for Linux, in particular the v4l and v4l2 teams, Gerd Knorr, and the Metzler Brothers among others.

Also, I would like to thank Marla, who has the grace to accept my imperfections and idiosyncrasies unconditionally, including my obsession with projects such as this.

1.5. Feedback

Please send any additions or comments pertaining to this document to the following email address: `<hshane[AT]austin.rr.com>`. In particular, if you have information about new devices or interfaces supported or errata, please contact me so we can keep this document up−to−date!

1.6. Conventions Used in this Document

The following conventions are used in this document and are outlined here for those who may not yet have a complete understanding of how to access and control the underlying operating system in Linux, which is usually via the Bash shell.

First, filenames are referenced in a paragraph like so: `/path/file`

Commands in Linux are executed (or 'called') at the command prompt, otherwise known as the 'command line.' If you are in the non−graphical (text−based) environment, you will usually be presented with the Bash shell prompt which is a dollar sign:

```
$
```

...or the hash mark:

```
#
```

...if you have logged in as root or have otherwise acquired root, or 'superuser' privileges. You can also access the Bash shell in the X window system, otherwise known as X or X11, with an xterm or similar X−terminal−emulator. Commands to be performed at the Bash prompt, but referenced in a paragraph of this document, usually look like this: **do this now**

Commands and/or the resulting output of commands may also be outlined with screen output in their own paragraph or heading:

```
$  date
Sun Jul 27 22:37:11 CDT 2003
```

When a command is written in front of the Bash prompt (e.g., **$ date** above), it is assumed the [Return] or [Enter] key has been pressed after the command, possibly followed by the output on a new line (e.g., as in the date in the above example).

2. Enabling Support for Your (Webcam) Hardware in Linux

2.1. Drivers and Modules

For your webcam to work you will need support for the connection and support for the actual camera hardware. Those who are already versed in kernels and modules and how to load them should skip to Section 2.2, which addresses support of the connection type. If you know your USB, IEEE 1394 or whatever bus you will be connecting your camera to is already configured and working, you should move on to the list of specific webcam hardware listed in Section 2.3.

Webcam drivers are usually available one of three ways: within the kernel, as a compilable stand alone module, or available as a pre−compiled (packaged) binary driver from your Linux distribution.

2.1.1. Module or In−Kernel?

As a rule, often the stock kernel, or working part of the operating system, of your initial installation may already have support for what you need. Your Linux distribution vendor has likely enabled the most common options already, including the bus, or connection type, and drivers for common camera models. The driver exists either as a loadable module or within the already running kernel.

An easy way to tell if the driver is enabled is to use the **dmesg** command piped into **less** (for easy paging) to look for an acknowledgement that it was loaded when your system started up:

```
  $  dmesg | less
```

...which may yield something like the following, depending on your hardware:

```
  Dec 18 17:35:18 localhost kernel: hub 5−0:1.0: USB hub found
  Dec 18 17:35:18 localhost kernel: hub 5−0:1.0: 2 ports detected
  Dec 18 17:35:18 localhost kernel: Linux video capture interface: v1.00
  Dec 18 17:35:18 localhost kernel: quickcam: QuickCam USB camera found (driver version QuickCam USB $Date: 2005/01/07 13:29:53 $)
  Dec 18 17:35:18 localhost kernel: quickcam: Kernel:2.6.7 bus:1 class:FF subclass:FF vendor:046D product:0840
  Dec 18 17:35:18 localhost kernel: quickcam: Sensor HDCS−1000/1100 detected
  Dec 18 17:35:18 localhost kernel: quickcam: Registered device: /dev/video0
  Dec 18 17:35:18 localhost kernel: usbcore: registered new driver quickcam
```

If you don't see it, the particular driver may exist as a loadable module. If you know what that module is named, try using `find`; in this example we are looking for the 'ibmcam' module:

```
 $  find /lib/modules −name ibmcam.o
```

Note that up until the 2.4 series modules had the suffix `.o`; for 2.6+ series kernels this was replaced with `.ko`.

You can get a list of all modules available by typing the following at the command line:

```
 $  ls −R /lib/modules/`uname −r`/kernel
```

Where **`uname −r`**, surrounded by forward tick marks, is your kernel version number. The following output is an example of what you might find in a USB webcam−ready kernel , where everything is loaded as a module (all but the relevant lines have been edited for brevity):

```
  ./usb: usbvideo.o usbcore.o ibmcam.o
```

Once you know which module your camera needs you can find out if it is already loaded by typing the following at the command line:

```
#  lsmod
```

As shown by the prompt above, you will need to have root privileges to do this. You should get output similar to the following:

```
cdrom                    29312   0  (autoclean) [sr_mod]
usb-ohci                 17888   0  (unused)
usbcore                  56768   0  [scanner ibmcam usbvideo usb-ohci]
ibmcam                   39680   0
```

Most stock kernels are compiled with `kmod`, which enabling automatic loading of necessary modules when the appropriate hardware is detected. It may not always do so, however, so if you don't have the particular module you're seeking loaded and you think the module may be available, try loading it manually with **modprobe**, as in the following using the `ibmcam` module as an example:

```
# modprobe -v ibmcam
```

Drivers for specific webcam models, or links to project pages hosting code for drivers, are outlined in Section 2.3. The drivers are usually available one of three ways: within the kernel, as a compilable stand alone module, or available as a pre−compiled binary from your Linux distribution.

If the support for your driver is not found either enabled statically within the kernel or as a module, don't despair. Drivers for numerous models are in the Linux kernel source (available directly from kernel.org source code repository), or in code offered separately from the kernel that can be configured to work with your current setup as oulined in Section 2.1.2. If your webcam driver is available in the kernel source but not enabled as a module or otherwise in your default system, you can either recompile the kernel from the source code you have or obtain a new version of the kernel source, either pre−packaged by your Linux distributor or directly from the previous link (as a so−called "vanilla" kernel). If you are unfamiliar with the prerequisites and procedure of compiling your own kernel, I direct you to the Kernel HOWTO for more information.

2.1.2. Patching, Source−Only or Precompiled Binary?

You may find that your webcam is supported by only a kernel patch, by a source−only driver not requiring a kernel recompile, or you may even be lucky enough to have a distribution that makes a pre−compiled and packaged binary driver available for your computer's architecture. The procedure involved in the former is largely beyond the scope of this document and is probably best outlined in the documentation available on the web page of your particular model's driver found in Section 2.3. Some further more general documentation on these processes are, however, addressed in Section 5

2.2. Supporting the Connection Type

2.2.1. USB Webcams

If you have a USB webcam, it is likely a Linux driver has been written for your device. There are two ways of supporting USB devices in Linux. One is the more traditional kernel support, and the other is through libusb. For at least one webcam category, the STV0680−based models, working libusb support is recommended, at least according to the Sourceforge page on the subject.

Unless you know your driver requires libusb support, you should probably stick with the more conventional in−kernel support for USB devices beginning in Section 2.2.1.2.

2.2.1.1. Libusb

Libusb is a library that allows access to the USB functions in Linux through userspace and without the need to enable kernel support and insert modules. Most distributions, at this point, are offering libusb in their stable branches (and some install it by default), so if you don't already have kernel support for USB devices, then you may only have to install the libusb package in order to access your device. You must have USB device filesystem support enabled in your kernel, which most distributions do. To find out for sure, issue the following at the command line:

```
$  cat /proc/filesystems
```

You should see (among others):

```
nodev    usbdevfs
nodev    usbfs
```

You may need to mount usbdevfs to enable it and see the device files, which you can do at the command line with **mount −t usbdevfs none /proc/bus/usb**. Don't try to use libusb while your particular *kernel* webcam support is enabled either statically or the module loaded; you can only use one at at time.

You can obtain the libusb package in `.rpm`, `.tgz` or `.deb` format from your Linux distribution.

2.2.1.2. Linux Kernel USB Support

Kernel support is required for USB webcam support if not using libusb (outlined above).

For 2.2 and 2.4 series kernels, your USB webcam may require the module `usbvideo` to function. This is not required in the 2.6+ series.

For generic USB bus support in Linux, you will need USB subsystem support in your kernel, whether usb−ohci, usb−ehci, or whatever flavor of USB driver your system prefers. USB subsystem support has been present in the Linux kernel since the late 2.2 series. For a more in−depth discussion of USB support in general, I direct you to the Linux−usb project site. If you want to find out which modules are loaded, at the command line or in an xterm, type the following:

```
#  lsmod
```

As shown by the prompt above, you will need to have root privileges to do this. You should get output similar to the following:

```
cdrom               29312   0  (autoclean) [sr_mod]
usb-ohci            17888   0  (unused)
usbcore             56768   0  [scanner ibmcam usbvideo usb-ohci]
ibmcam              39680   0
```

If you don't have the particular module you're seeking loaded and you think the module may be available, try loading it directly (using the usb ibmcam module as an example):

```
#  modprobe -v ibmcam
```

...at which point you should see something like the following:

```
Using /lib/modules/2.4.20/kernel/drivers/usb/ibmcam.o
```

By placing the entry `ibmcam` (for example) in `/etc/modules` (note that this varies by distribution), you can have the module load at boot−time automatically. You can then confirm the module was loaded by checking the syslog or in the boot−time record with **dmesg | less**), where you should see an entry such as the following:

```
Oct 18 12:43:12 K7 kernel: hub.c: new USB device 00:02.3−2, assigned address 3
Oct 18 12:43:12 K7 kernel: ibmcam.c: IBM PC Camera USB camera found (model 2, rev. 0x030a)
Oct 18 12:43:12 K7 kernel: usbvideo.c: ibmcam on /dev/video1: canvas=352x240 videosize=352x240
```

2.2.2. IEEE 1394 (Firewire", i.Link")

IEEE 1394 webcams require an IEEE 1394 PCI card or an IEEE 1394 bus port on your mainboard. The IEEE interface has been supported in Linux since the early 2.4−series kernel. If you are lucky enough to own such a device, generic information on support of the IEEE 1394 bus in Linux can be found at www.linux1394.org. If you have a kernel older than 2.4.2, you will need to patch your kernel with one of the patches found on this page matched to your kernel version. In addition, you will require libraw1394. The previously referenced linux1394.org site has a great installation guide.

The IEEE1394 Digital Camera List, by Damien Douxchamps, offers an outstanding summary of the capabilities of IEEE 1394 cameras as well as the current status of support for individual models.

2.2.3. Generic Parallel Port Support for Parport Webcams

For 2.2 and 2.4 kernel systems, parallel−port support must be enabled statically or as a module (stock kernels usually have this enabled by default). You may want to read more generic info about parallel−port device support under the Linux kernel before starting this process. To find out for sure if the module `parport` is loaded, you can check the `dmesg` file or use **lsmod** as outlined above. Using **dmesg | less**, you should see (among many other lines) the following:

```
Mar  3 08:00:25 K7 kernel: parport0: PC−style at 0x378 (0x778) [PCSPP,TRISTATE]
Mar  3 08:00:25 K7 kernel: parport0: irq 7 detected
```

If you are compiling your own kernel, enable 'Parallel Port support'. You should enable 'IEEE 1284 transfer modes', and if you have x86 type architecture, you should also enable 'PC−style hardware'.

If modprobe returns an error when you attempt to load the module, note that you may need to determine and supply the hardware address when invoking **modprobe**. The most common address is 0x378 for an x86 system; 0x278 and 0x3BC are other possibilities for integrated or ISA parallel ports. Add−in PCI parallel ports may have unusual base addresses. You can also arrange multiple devices with either the parport_pc or parport_arc modules, though that topic is beyond the scope of this document.

WARNING: Be sure you have the correct address before entering this information at the command line or else your machine may become unstable, crash or otherwise implode.

Your parallel port should be set to preferably "EPP" mode, or alternatively ECP/EPP. "Bidirectional" (also known as "BPP" or "PS/2") may work, albeit much more slowly. "Unidirectional" mode is unsuitable for scanning. The above setting can usually be accessed through your BIOS menu, at least on x86 systems.

2.3. Specific Webcam Models

Note that this information is frequently changing. The Linux−USB Device Overview site is a great place to look if you have a USB webcam. Also, you will want to check for your model's homepage at http://www.exploits.org/v4l/. The information compiled below on specific webcam models is from the same source, so you may find more up−to−date information through the previous link. If you can't find an entry for your particular hardware, you can find links to resources on how to write your own driver!

It is important to note that if your camera isn't listed, the easiest way to find out if your camera is supported is to find out what chipset is used in its manufacture.
This information is usually present in the specifications published in your webcam's manual or on the manufacturer's website.

If you can't find your camera model listed and aren't sure what chipset your camera is made with, you should consider searching and/or subscribing to the video4linux−list mailing list hosted by Redhat.

2.3.1. 3com HomeConnect PC Digital Webcam

This driver is supported with the kernel patch located at the homeconnectusb project web page. It may require a kernel recompile after patching depending on your kernel version.

2.3.2. CPiA based Webcams

Please see the project home page for up−to−date information. This chipset has been used in the manufacture of both USB and parallel port webcams including the following:

- Aiptek HyperVcam Fun USB (non−OV511 based)
- Creative Video Blaster WebCam II USB and parallel−port
- CVideo−Mail Express parallel−port
- Digicom Galileo USB and Digicom Galileo Plus
- Dynalink Digital Camera
- Ezonics EZCam (not Pro or Plus)
- I−View NetView NV200M
- Microtek EyeStar USB
- Pace Color Video Camera USB
- SuperCam WonderEye
- TCE Netcam 310 USB
- Terracam USB (non−OV511 based or Terracam Pro)
- Trust SpaceC@m Lite USB and SpaceC@m 100
- Utopia USB Camera
- ZoomCam USB and parallel−port

2.3.3. SE401, SE402 and EP800 based USB webcams

This project is a work in progress. The drivers and other useful information are available at the project homepage located here. As of writing this, it is necessary to patch and recompile your kernel in order to obtain support for these models. The driver supports the following:

SE401 chipset via the 'se401' driver:

- Aox SE401 camera
- Philips PCVC665 USB VGA webcam 'Vesta Fun'
- Kensington VideoCAM PC Camera (Models 67014−67017)

SE402 and EP 800 chipsets via the 'epcam' driver

- Spypen Actor
- Rimax Slim Multicam
- Concord Eye−Q Easy
- Creative PD1001
- Chicony DC−100
- Endpoints SE402 and EP800

2.3.4. OmniVision based Webcams

This category includes a multitude of webcam and video−capture devices manufactured by Omnivision, including the OV511(+), OV518(+), OV6620, OV6630, OV7610, and OV7620AE. The project homepage is here. Supported models include:

- Aiptek HyperVcam Home and Mobile
- Amitech AWK−300
- I−view NetView NV300M
- TEVion MD9308
- Intel Me2Cam
- Dlink DSB C100, C300
- Hawking Tech. UC−110, UC−300 and UC−310
- Puretek PT−6007
- Alpha Vision Tech AlphaCam SE model AC−520
- Creative Labs WebCam model PD1001 with OV518 chipset
- Creative Labs WebCam 3, WebCam Go, Webcam Go Plus
- Elecom UCAM−C1C20
- Elta WEBCam 8211 PCC
- Ezonics EZPhone Cam
- Philips ToUCam XS (old version with OV518)
- LG Electronics LPC−UM10
- Lifeview various USB Life TV models
- Genius VideoCam Express
- AverMedia Intercam Elite
- Maxxtro Cam22U
- MediaForte MV300, PC Vision 300
- Terratec TerraCam PRO and some TerraCam models
- OmniVision (except those with OV519)
- TRENDNet TV−PC301
- Trust Sp@ceC@m USB
- Lifetec LT9388
- BestBuy EasyCam U
- Maxell Maxcam
- TCE NetCam 310u
- Medion MD9388
- Webeye 2000B
- Suma eON
- Prochips PCA−3100
- Ezonics EZ USB Cam II (the OV511+ models)
- Waytech I−Pac VIC−30

* Zoom Telephonics ZoomCam III USB (model 1598)

2.3.5. Logitech (formerly Connectix) Quickcam Support

The QuickCam VC USB and parallel port model webcams are supported by the driver offered here. A kernel patch and recompile are necessary for support of this model.

The Quickcam driver is represented by two different projects that offer two different flavors of driver for certain Quickcam models, both of which are stand−alone drivers that do not require a kernel patch or recompile. The qce−ga and qc−usb drivers support the following models:

* Logitech (earlier models of) Quickcam Express
* Quickcam Web
* Legocam
* Dexxa Webcam
* Labtec Webcam

The qc−usb driver is more experimental but reportedly works better on some models such as the Quickcam Web. Also, I have recieved correspondence that newer versions of the Logitech Quickcam Express no longer work with the above drivers; instead this page offers an experimental driver that claims to support the newer model.

Note to Redhat users: The qce−ga driver doesn't compile properly using the modified kernel source provided in Redhat 9, but a fix is available here.

Some Logitech camera models are supported by the Philips driver in Section 2.3.8.

2.3.6. ICM532 Based Webcams

One driver for this chipset, homepage here, is now merged into the 2.6 kernel source; the other is (per the developer's own description) experimental and available here. Either or both claim to support the following models:

* IC−Media Corp Pencam
* Newer versions of the Logitech Quickcam Express
* Newer versions of the Labtec Webcam
* Biolux 654 microscope
* Ezonics EZCam USB II (uvt8532)
* Ezonics EZCam USB III
* TerraCam USB
* Stick Webcam
* Mini WebCam
* Tucan PenCam
* Che−ez! Webbie
* SNAKE EYE SI−8480/8481
* PC CAM CP03
* WEB Camera PBC0006
* Clipcam

2.3.7. NW802 Based Webcams

This chipset, manufactured by DIVIO, is supported by the driver found here. The models supported include the following:

- BTC SurfCam CMOS300k
- Mustek WCam 300
- Logitech QuickCam Pro USB (the earlier "dark focus ring" model)

2.3.8. Philips USB Webcams

Because of the expiration of the Non−Disclosure−Agreement between Philips Corporation and the former maintainer of the `pwc` driver, the previous kernel support for Philips PWC−chip−based webcams has been removed. Luckily a new, still experimental driver that does not require a proprietary module is under development. The old site, with a discussion of the change, can be seen at http://www.smcc.demon.nl/webcam/; the new driver is maintained at saillard.org with more information at the PWC Documentation Project.

Philips models supported by the above include the following.

- PCA645VC
- PCA646VC
- PCVC675K Vesta, Vesta Pro and Vesta Scan
- PCVC720K/40 ToUCam XS, ToUCam Fun, ToUCam Pro and ToUCam Scan
- Askey VC010
- Creative Labs Webcam 5, Pro Ex
- Logitech 3000 and 4000 Pro, Notebook Pro, and Zoom
- Samsung MPC−C10 and MPC−C30
- Sotec Afina Eye
- Visionite VCS UM100 and UC300

2.3.9. SPCA50X USB Camera Linux Driver

Information regarding this chipset can be found here, and is under heavy development and includes partial or complete support for the following models:

- Kodak DVC−325 and EZ200
- Creative PC−CAM 300, 600, 750
- Genius VideoCAM Express V2
- Micro Innovation IC 200/IC 150
- Logitech ClickSmart 310, 420, 510, 820 and Cordless models
- Logitech Pocket750
- Benq DC 1016, 1300, 1500, 3410
- Flexcam 100
- Aiptek MegaCam, [1.3 Megapixel] Mini PenCam and PocketCam 1.3M Smart
- Finet Technology Palmpix DC−85
- Pure DigitalDakota
- 3Com Home Connect lite
- Megapix V4
- Mustek gSmart: Mini, Mini2, Mini3, LCD 2, LCD 3

- Digital Dream Enigma 1.3, Epsilon 1.3
- Maxwell Compact Pc PM3
- Jenoptik models
- Minton S−Cam F5
- D−Link DSC−350
- Trust FamilyC@m 300 Movie
- Aiptek Pocket DV, PocketDVII, DV3100+, mini PenCam 2, PocketCam 3M, Pencam SD 2, Pocket DV3500
- Hama Sightcam 100
- Micro Innovations IC50C, IC400c
- FlyCam USB100
- Arowana USB Camera 300 K
- Intel Easy PC Camera, CS120 (Easy PC Share), PC Camera Pro (CS431), Pocket PC Camera (CS630)
- Grandtec V.cap
- Sigma−Apo Petcam

2.3.10. STV0680 based Models

The USB version of webcams made with this chipset are supported by the 2.4.18 and above kernel with the `stv680.o` module. Alternatively, you can obtain the source from the project homepage. This driver supports models including the Aiptek Pencam and the Nisis Quickpix 2.

If you have a serial version, the main one of which is the Scan e−Studio, you should go here.

2.3.11. Winbond w9966cf

This is a driver for the parallel−port interface that supports the Philips SAA7111 CCD−control chip as found on the Lifeview Flycam SUPRA webcam. It is included in the late 2.4 kernel series and later under the heading 'video4linux' support. The homepage for this project is here.

2.3.12. Xirlink C−it" HDCS−1000 based Webcams

This driver is for the USB webcams manufactured by Xirlink, IBM (PC Camera) and Veo Stingray model webcams. Support has been in the Linux kernel USB section since 2.2.12. The homepage is at http://www.linux−usb.org/ibmcam.

3. Accessing the Video Device

The following section applies to all connection types.

3.1. The Video Devices Node

The Linux kernel requires a virtual device node be created to access and control the device in question. It may have already been created for you at boot−up; **ls −l /dev/video*** (with an asterisk) or alternatively **find /dev −name video*** or even visual inspection of the `/dev` directory with your favorite file manager can give you an idea if the video devices exist. If so you can proceed to Section 3.2; if not you will need to create them manually.

An easy way to create them, if available with your Linux distribution, is use of the `MAKEDEV` script, which may be located in `/dev` or the usual places for storing executable commands (`/bin`,`/sbin` and so on). The manual page for `MAKEDEV` (**man MAKEDEV**) can guide you further, but be aware of the device−specific command options. If `MAKEDEV` doesn't work or doesn't exist, or you just prefer doing things the hard way, move on to the next paragraph.

A device can be created as a block (such as a drive), a fifo (file−in−file−out or pipe, as in xconsole) or a character device, which represents other hardware. Each device has a major and a minor number "coordinate" to tell the kernel what it is and where to access it. These numbers are not arbitrary. The major number 81 with minor number 0, 1, 2, and so on are by convention assigned to Video4linux devices, including TV tuner boards and webcams. In order to create the video device `/dev/video0`, use **mknod** at the command line:

```
#  mknod /dev/video0 c 81 0
```

where **c** represents a character device.

You can use the following script, which I have borrowed from the kernel source (located in `linux/Documentation/video4linux/bttv/MAKEDEV` of the source tree):

```
#!/bin/bash
function makedev () {
     for dev in 0 1 2 3; do echo "/dev/$1$dev:
     char 81 $[ $2 + $dev ]" rm −f /dev/$1$dev
     mknod /dev/$1$dev c 81 $[ $2 + $dev ] chmod
     666 /dev/$1$dev
     done

     # symlink for default device
     rm −f /dev/$1 ln −s /dev/${1}0 /dev/$1
     }

     # see http://roadrunner.swansea.uk.linux.org/v4lapi.shtml
     echo "*** new device names ***" makedev video
     0 makedev radio 64 makedev vtx 192 makedev vbi 224
     # "*** old device names (for compatibility only) ***"
     #makedev bttv 0 #makedev bttv−fm 64 #makedev bttv−vbi 224
```

Simply copy and paste the above into your favorite editing program, save it as MAKEDEV or whatever name you like, make it executable (i.e., **chmod u+x MAKEDEV**), and then execute it as root:

```
#  ./MAKEDEV
```

3.2. Groups and Permissions

It is a good idea to be sure that your user account can access the device once all modules are loaded and device nodes created. The most security−conscious way to do that is to add access for a particular group. On my system, the members of the group 'video' are allowed to use the webcam, scanner and other photographic devices. The way to accomplish this is to first change the ownership of the devices in `/dev` like so (as root):

```
#  chown root.video /dev/usb/video*
```

...where **root.video** are the owner and group the device will now belong to. Obviously, the specific command will vary by your system and the type of device. It is important that you change the ownership of the device node itself and not the symlink; symlinks' ownerships are affected only by changing the parent devices or files they point to.

To see if your user account is a member of the group in question, as root issue the following command: **grep −e video /etc/group**. You should see something like the following:

```
video:x:44:
```

...where '44' is the group number. Since no members follow the last colon in the 'video' group, we can add them, let's say user 'jhs' with the command

```
#  adduser jhs  video
```

After this, it's simply a matter of allowing read and write access for the user in question of the device like so:

```
#  chmod g+rw /dev/v4l/video0
```

...where **g+rw** means add **r**ead and **w**rite access for **g**roup. See the documentation for chmod (**man chmod** or **info chmod**) for further info.

4. Framegrabbing Applications

4.1. Command Line Programs

As implied by the title these do not require the X Window System to operate your camera and capture images.

4.1.1. Streamer

Streamer is a versatile program that allows a capture from a webcam or video device using only the command line. It may be offered in your Linux distribution's Xawtv package, or may need to be fetched separately as in Debian. You can find it and more information at Gerd Knorr's Xawtv homepage.

To take a standard JPEG picture from the command line where the camera is accessed through `/dev/video0`:

```
$  streamer −c /dev/video0 −b 16 −o outfile.jpeg
```

...where **−b** is the number of colors (in bpp, whether 15, 16, 24 or 32) and **−o** is the output filename that will be dropped into the current directory (specify **−o** `/path/outfile.jpg` to place it elsewhere). If you are going to capture multiple images be sure to append the output file name with zeros, as streamer can name the capture files in sequence, i.e., **−o** `outfile000.jpeg` becomes `outfile001.jpeg`, `outfile002.jpeg`, and so on.

To make an .avi file:

```
$  streamer −q −c /dev/video0 −f rgb24 −r 3 −t 00:30:00 −o /home/jhs/outfile.avi
```

...where **−q** is for 'quiet' execution (no message output), **−f** is 'format' (rgb24 is TrueColor avi), **−r** is the frames per second and **−t** is the time of recording (30 minutes). Streamer can capture raw and Quicktime" (non−Sorensen) formats and can capture audio as well. See **streamer −−help** for more information.

4.1.2. camE

CamE is a command−line program that works in daemon mode to capture frames from your v4l device for archive or upload (to a webserver, for example) via ftp or scp. You can overlay other graphics, timestamp the frames, or add other dynamic text all by altering the appropriate line in the configuration file. See the camE homepage for more information.

4.1.3. Motion

Motion is a brilliant program that is able to monitor the video signal from one or several webcams. It can record periodic snapshots, and when motion is detected, record an mpeg and/or perform another action such as sending an email or executing a command. It can track and graphically mark the motion it detects, feed files via an http server to your website, stream them to another application and more. The number of command line options may be intimidating; there is however, a Wiki available online that outlines the various command and configuration file options nicely. The motion homepage can be found here.

4.1.4. Webcam

Webcam is an automated command line tool for operating a webcam that is also available from the Xawtv homepage. It is excellent for automated operation such as from a cron job, as it requires no command line options, only a previously edited configuration file (usually `~/.webcamrc`). It is similar to camE above in that one can captures images and upload them to a Web−Server via ftp or ssh.

4.1.5. SANE

SANE, or Scanner Access Now Easy, supports access of v4l devices including webcams in later versions. If you are familiar with using a photographic scanner device in Linux, you may be interested in using SANE for image capture, especially since a few devices double as both scanners and digital cameras. See the relevant sections of the Scanner−HOWTO here.

4.2. GUI(Grapical User Interface)−Based Programs

4.2.1. Xawtv

Xawtv is an X−based program for accessing video devices in Linux including TV tuning devices and webcams. The home page is at http://bytesex.org/xawtv.

When you first try out your webcam, and you think things are configured right, use the −hwscan option:

```
$  xawtv -hwscan This is xawtv-3.72, running on Linux/i686 (2.4.21) looking for available devices

/dev/v4l/video0: OK          [ -device /dev/v4l/video0 ] type : v4l name : BT878(Hauppauge (bt878)) flags: overlay capture tuner
/dev/v4l/video1: OK          [ -device /dev/v4l/video1 ] type : v4l name : IBM USB Camera            flags: capture
```

...so now you can see the available devices (your output may differ substantially). Try opening an xterm and running xawtv, grabbing from your webcam video device:

```
$  xawtv -c /dev/video1 This is xawtv-3.72, running on Linux/i686 (2.4.21)
```

...and (hopefully) your camera will begin capturing to a window on your desktop. You may see some error messages in your xterm if things don't work that can be helpful to diagnose configuration problems. If you aren't interested in all that, and things work for you, launch from your window manager's menu next time. You can read about more xawtv options with **man xawtv**.

4.2.2. Gqcam

Gqcam is a graphical GTK+−based application originally written to access Connectix QuickCams but now supports nearly all Video4Linux compatible webcam devices. It has an intuitive interface that makes viewing, taking snapshots, and configuring webcam settings blissfully easy. It is highly recommended for those who only want to take a picture here and there without editing a configuration file or using the command line.

4.2.3. Camorama

Camorama is a graphical GTK+2.0−based application very similar to gqcam written for the Gnome2 desktop. The home page is here.

4.2.4. GnomeMeeting

GnomeMeeting is a VOIP/IP−Telephony application for the Gnome2 desktop that also supports videoconferencing with a webcam. The homepage can be found at gnomemeeting.org.

5. Troubleshooting

5.1. Help, I have a USB webcam and don't know exactly what model it is and/or who the manufacturer is. What do I do?

Use **lsusb**; it can give you an idea of what other USB devices are available on your system, too:

```
$  lsusb
Bus 007 Device 001: ID 0000:0000
Bus 006 Device 001: ID 0000:0000
Bus 005 Device 001: ID 0000:0000
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 003: ID 0545:8080 Xirlink, Inc. IBM C-It WebCam
Bus 003 Device 002: ID 046d:0840 Logitech, Inc. QuickCam Express
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 003: ID 051d:0002 American Power Conversion Back-UPS Pro 500/1000/1500
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
```

The numbers after 'ID' are the Vendor and Product numbers, respectively. They can then be looked up in the Linux USB ID catalog.

If **lsusb** is not available to you, and you have support for `/proc` filesystem support and USB−filesystem support, issue the following at the command line:

```
$  cat /proc/bus/usb/devices
```

You should receive output including (but not necessarily limited to) the following:

```
T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=01 Dev#=3 Spd=12 MxCh=0
D:  Ver= 1.01 Cls=ff(vend.) Sub=ff Prot=ff MxPS=8 #Cfgs=1
P:  Vendor=0545 ProdID=8080 Rev= 3.0a
S:  Product=USB IMAGING DEVICE
```

The line beginning "T:" is the USB bus the device is attached to. The "P:" indicates (obviously) the vendor and product ID, which are catalogued at the linux USB Project homepage.

5.2. Help, I can't find the camera device in `/dev`!

Assuming your connection type is supported, and your camera is working, see Section 3.1.

5.3. Help, I can see the camera device (both in person and as a device node in `/dev`), but I can't access it!

See Section 3.2.

5.4. Help, my camera has a driver that is source−only, i.e., has to be built by me! Where do I start?

First, check if your Linux distribution offers a pre−compiled binary of the driver. You can then load it as you normally would for a module. If that is not the case, be sure you have kernel sources installed. You will also need at a minimum GNU make, gcc, binutils and perhaps other programs installed depending on your distribution. (Debian users should see the next section for instructions specific to that distribution.)

Download the driver source (in this example named `src.tar.gz`) and uncompress/untar it:

```
$  tar -xvzf src.tar.gz
```

Then, change to the directory of your kernel source:

```
#  cd /usr/src/linux
```

Make the necessary source files:

```
#  make oldconfig # make dep
```

Now, change to the directory where you unpacked the driver source and read the README and/or INSTALL files for instructions on how to make the driver. Usually this involves some combination of "make" "make all" and/or "make install." Assuming it compiles correctly, you can simply load the new module with **modprobe**. If you have any problems, see Section 5.7.

5.5. I am using Debian GNU/Linux. Is there an easier way to go through all this kernel compiling stuff and building of source modules?

It is far simpler, in your author's humble opinion, to use the automated `kernel−package` utility. First, install it and the dependencies using **apt−get**. Next, install the kernel source that you want (e.g., **apt−get install kernel−source−2.X.X**). Untar the bzip2'd kernel source with **tar −xvjf** and then make a symbolic link called `linux` that points to the new source:

```
#  ln -s /usr/src/kernel-source-2.X.X /usr/src/linux
```

Then **cd /usr/src/linux** and clean: **make−kpkg clean**, followed by **make menuconfig** or **make xconfig** as you would if compiling a new kernel. Next, you can use **make−kpkg kernel_image** and then install your new kernel package that has been deposited in `/usr/src` with **dpkg −i ../kernel−image−2.X.X**. Next, you can get the pre−packaged source driver using `apt`. So, in the case of the Quickcam Express, the package is `qc−usb−source`:

```
#  apt-get install qc-usb-source
```

...and untar the archive:

```
#  tar -xvzf qc-usb-modules.tar.gz
```

This will uncompress the source into the `/usr/src/modules` directory. The final step, while still in `/usr/src/linux` is to make the modules with kernel−package:

```
#  make-kpkg modules_image
```

Install the new package, in `/usr/src/`, called `qc-usb-modules-[arch].deb` using **dpkg –i**. Finally, load the module:

```
#  modprobe quickcam
```

Check the documentation in /usr/share/doc/kernel–package for any problems.

5.6. Help, my camera is supported by a driver that has to be patched into my kernel! What do I do?

See the section on patching of the Kernel–HOWTO at The Linux Documentation Project.

The short and unguaranteed version of patching, by your humble author, goes as follows: Be sure you have the same prerequisites outlined in Section 5.4 installed. First, on the command line or in an xterm change to the source directory of the kernel version you are (or will be) running with the camera patch (in this example named `patch.diff`).

```
#  cd /usr/src/linux
#  patch -p1 -E patch.diff
```

You should see a confirmation that the 'hunks' were successfully applied. At this point, you can **make menuconfig** or whatever program you use to recompile, enabling the appropriate support. If any of the hunks failed, or you run into any problems in addition to the link referenced above you, should consult **man patch** and Section 5.7.

5.7. Help––as in, where can I get more of it?

See the video4linux mailing list headquarters at https://listman.redhat.com/mailman/listinfo/video4linux–list.

5.8. Help, I want to contribute to Video4Linux support in Linux! Who do I get in touch with?

See this page.

A. Gnu Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft cense designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works in which its purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world−wide, royalty−free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee and is addressed as "you". You accept the license if you copy, modify, or distribute the work in a way, thus requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front−matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical, or political position regarding them.

The "Invariant Sections" are certain Secondary Sections in which titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary, then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections, then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front−Cover Texts or Back−Cover Texts, in the notice that says that the Document is released under this License. A Front−Cover Text may be at most five words, and a Back−Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine−readable copy, represented in a format where its specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format in which its markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard−conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF, and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine−generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats, which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document in which its title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section, when you modify the Document, means that it remains a section "Entitled XYZ", according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front−Cover Texts on the front cover, and Back−Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine−readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer−network location from which the general network−using public has access to download using public−standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers. If the Modified Version includes new front−matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles. You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties−−for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front−Cover Text, and a passage of up to 25 words as a Back−Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front−Cover Text and one of Back−Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection and distribute it individually under this License, provided you insert a copy of this License into the extracted document and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise, they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense, or distribute the Document is void and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute, and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front−Cover Texts, and no Back−Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front−Cover Texts and Back−Cover Texts, replace the "with...Texts." line with this: with the Invariant Sections being LIST THEIR TITLES, with the Front−Cover Texts being LIST, and with the Back−Cover Texts being LIST. If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.