The Scanner HOWTO

Howard Shane

<hshane[AT]austin.rr.com>

Revision History

Revision 1.1	2004−05−16	Revised by: jhs
Libusb and kernel 2.6−series updates, clarifications		
Revision 1.05	2004−01−15	Revised by: jhs
Miscellaneous errata and updates		
Revision 1.0	2003−08−19	Revised by: tm
Initial release, reviewed by LDP		
Revision 0.04	07−03	Revised by: jhs
Clarified, revised and edited after inviting feedback from participants of the SANE−devel mailing list		
Revision 0.01	06−03	Revised by: jhs
Completed draft.		

This document was written to document the steps necessary for access and use of a photographic scanner device on a system running Linux.

Table of Contents

Table of Contents

1. Introduction

This document was written to assist the Linux user in setting up a raster image scanner device, including flatbed, hand−held, video− and still−cameras, frame−grabbers and so on. It does not address how to use the available software tools to achieve a particular photographic result or to utilize your scanner device's features to the fullest extent. For that information please consult the application home pages referenced in the text and the manufacturer's information that accompanied your hardware.

Finally, this document does not answer the question "What type of scanner should I buy?" The answer varies depending on what you are looking for in a scanner device. I suggest looking at the supported hardware list link in Section 2 and also this link within the SANE−project FAQ.

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 system failure, destruction of your hardware and the loss of your irreplaceable data. Proceed with caution and be aware that although errors are unlikely, the author can nonetheless accept no responsibility whatsoever for them.

All copyrights are held by their 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 initial release.

The latest version of this document can be found here.

1.4. Credits

I would like to thank Oliver Rauch, Henning Meier−Geinitz, Jonathan Buzzard, Laurent−jan, Jochen Eisinger and others who participate in SANE development and/or contribute to the SANE−devel mailing list, without whose input this project would have been difficult if not impossible to perform with any measure of quality−control. I would also like to thank the many individuals who have taken the time to email me new information and corrections.

Also I would like to thank Marla for graciously tolerating all the time I've spent banging on the keyboard working on projects such as this. You're the greatest.

1.5. Feedback

Please send any additions or comments pertaining to this document to the following email address : `<hshane[AT]austin.rr.com>`. As this is the first release I am particularly interested in any errata, so don't hesitate to contact me if you know of something I have wrong or needing updating. Also let me know if you know of any shortcuts, tools or bits of information that may help hapless users that you think should be included. I apologize in advance, but I cannot answer any technical questions or "plz help me" pleas regarding scanners; any sent my way will be forwarded to `/dev/null`; for sources of assistance including live help see Section 7.5 instead, but only after reading the relevant sections of this document in their entirety. I am neither an expert on scanners nor do I have every model of scanner ever manufactured available for testing. My only contribution to scanner support within Linux is the compiling of my own limited experience with the exhaustive input of others to produce a succinct but (hopefully) straightforward HOWTO.

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 almost always 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 the Bash shell prompt which is a dollar sign:

```
$
```

...or the hash mark:

```
#
```

...if you have logged in as root or have 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 depressed after the command, possibly followed by the output (e.g., the date).

2. General Support and Interface Type

There are four predominant types of scanner interfaces available and discussed in this document: SCSI, USB, parallel port, IEEE 1394. Linux support exists for most scanners as pioneered by the SANE project. This is not the same thing as TWAIN, which you may be familiar with if you have used a scanner device under another operating system such as Microsoft Windows". The latter protocol weds driver and user interface in a way that does not allow its use outside of that proprietary graphical environment. Thus SANE, or Scanner Access Now Easy, was conceived for use under (but is by no means limited to) the Un*x environment. The SANE standard allows for modularity where driver meets application and allows for much greater flexibility and portability. With SANE you can scan with your device using only the command line, you can design your own front−end application to use the SANE backend(s), access your scanner(s) over a network or even access your cameras and other video4linux devices to acquire photographs. As such SANE is SANE where TWAIN is not.

NOTE: Before reading any further you should check the SANE homepage at http://www.sane−project.org/sane−mfgs.html to see if your scanner device is supported. Alternatively you can use the sane supported scanners search engine.

If you have an integrated device, i.e., one that functions as a scanner, printer and/or fax, you can follow the steps below for the scanner functions using the appropriate interface just like a standard scanner. Those who own an HP officejet should consult the HP Officejet Linux Driver project site, which goes into excellent detail on how to get the various functions of this integrated device to work within Linux.

2.1. SCSI Scanners

These scanners are managed by an SCSI controller. In general, just about any scanner using an SCSI interface should work assuming the SCSI hardware is supported. You should check the SCSI controller list of the Hardware HOWTO if you are unsure if the SCSI controller is supported. If your SCSI controller came bundled with your scanner there is a chance your hardware may not be supported or is only partly supported, as the accompanying SCSI card may not function as a complete SCSI controller.

You should consult **man sane−scsi**, if you run into difficulty configuring your SCSI scanner at any point.

2.2. USB Scanners

You probably already know what a Universal Serial Bus (USB) connector looks like and where it plugs in. If you have a USB scanner your hardware is likely to be supported in Linux. Information on enabling the USB subsystem and USB scanner support is found in Section 2.5.

2.3. Parallel Port Scanners

Parallel−port scanners on the whole can be made to work if there is a backend that supports them, however if your device also has a USB port (which the vast majority of new scanners released nowadays do) and a working USB backend you are strongly encouraged to use that instead, as it may be more easily configured.

If your model has only a parallel−port interface and a proprietary or non−standard controller you could be out of luck. If you have found there is a supported backend for the parallel−port interface of your scanner, then you should see Section 2.8.

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

Some IEEE 1394 scanners are supported as of the time of this writing, particularly those manufactured by Nikon and Epson. The IEEE 1394 interface has been supported since the 2.4−series Linux kernel. IEEE 1394 scanners require your system be equipped with a IEEE 1394 PCI card or a mainboard IEEE 1394 port, as well as have IEEE 1394 support enabled in your kernel or as a loaded module. You should check the SANE supported devices by manufacturer link in Section 2 and read the manpage next to your hardware (if any) for any issues related to your specific hardware.

2.5. Operating System Support

If you don't have a USB scanner you should skip to Section 2.7, and if your equipment is of the parallel port variety you should go to Section 2.8.

2.6. USB Scanners and Libusb

This section was at one time entitled "USB Scanner Kernel Support," but the existence of libusb promises to make the need for a USB−scanner enabled kernel unnecessary. Libusb is a project to create a userspace (i.e., non−kernel) library to access USB devices regardless of operating system. For more information on the differences between these consult **man sane−usb**.

If you would prefer the more conventional kernel support for your USB Scanner, go on to Section 2.7.2, but be advised that kernel support for USB scanner devices is dropped in favor of libusb in kernel version 2.6.0 and higher. 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 scanner 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 *kernel* scanner 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. If you are planning on compiling your own SANE binary from source with libusb support enabled you will require the `libusb−dev` package as well.

2.7. Linux Kernel Support of your Scanner Device

Kernel support is required for SCSI, USB and parallel−port generic interface support and USB scanner support (if not using libusb). Your stock kernel may already have support for what you need, and the way to tell is to use the **dmesg** command and look for an acknowledgement that the driver in question loaded at bootup. If you don't see it, the driver may be present (but not necessarily loaded) as a module. To find out you can type the following at the command line:

```
 $  ls −R /lib/modules/X.XX/kernel/drivers
```

Where 'X.XX' is your kernel version number. The following output is an example of what you would find in a USB scanner−enabled kernel (though all but the relevant lines have been edited for brevity):

```
./usb:
scanner.o
usbcore.o
```

(A hint for newbies: if the info in dmesg or the above module list scrolls by too fast, you might try piping the output into 'less' (or 'more' if you don't have less): ***ls −R /lib/modules/X.XX/kernel/drivers | less*** *or alternatively catching it in a file:* ***ls −R /lib/modules/X.XX/kernel/drivers > file.txt****, where 'file.txt' will contain the info that can then be accessed with* ***cat [file] | less****.)*

The following information is arranged on the basis of scanner interface type. If your kernel doesn't contain the necessary support, you can always recompile your kernel. If you are unfamiliar with the process of compiling your own kernel, I direct you to the Kernel HOWTO for more information.

2.7.1. Kernel SCSI Support

If you have an SCSI−type interface, when invoking **make config**, **make menuconfig** or **make xconfig** etc., be aware that in addition to the option to support your particular SCSI adapter, generic SCSI device support is also required. Such generic devices are usually named **/dev/sg0, /dev/sg1**.... Since you probably already know if your card is supported from the supported SCSI controllers list, all that is required after confirming that your kernel supports your hardware and generic SCSI devices is to load the appropriate module(s):

```
# modprobe CARD_MODULE_NAME
```

```
# modprobe sg
```

...as root. Note there have been reports of ide−scsi emulation support (used for ATAPI−eide CDRW support) causing problems for scanner access; if you know your hardware is supported and you can't get things to work try unloading the ide−scsi module:

```
rmmod ide−scsi
```

...though it has been reported to me that this has been fixed as of recent (2.4.20+) kernels.

2.7.2. Kernel USB and USB Scanner Support

For USB scanner support, you will need USB subsystem support in your kernel, whether usb−ohci, usb−ehci, or whatever protocol of USB driver your system prefers. USB 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 have a 2.4−series of kernel or earlier and wish to use the kernel USB−scanner support to access your scanner (instead of libusb outlined in Section 2.6) you will need to have 'USB scanner support' enabled, which, if present, is visible within **dmesg**, or by **lsmod** if a loaded module. 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 including (but not limited to) the following:

```
cdrom                  29312   0  (autoclean) [sr_mod]
usb-ohci               17888   0  (unused)
usbcore                56768   0  [scanner ibmcam usbvideo usb-ohci]
scanner                 8704   0
```

If you don't have `scanner` loaded, and you know you have USB scanner support in your kernel as a module, try loading it directly:

```
# modprobe -v scanner
```

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

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

By placing the entry `scanner` 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:

```
May 16 23:17:25 K7 kernel: usb.c: registered new driver usbscanner
May 16 23:17:25 K7 kernel: scanner.c: 0.4.6:USB Scanner Driver
```

2.8. Parallel Port Scanners

By now you've probably figured out that configuration of parallel port scanners may be problematic. Again, if your device has both a parallel port interface and a USB interface you should consider selecting USB to make the setup process easier.

2.8.1. Kernel Parport Support

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. One 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.

Depending on whether your parallel port scanner requires SCSI support, you may need to patch your kernel for parport−SCSI support. You can find that suite of tools at www.torque.net/parport/ppscsi.html. If this is required you will also need to enable the following:

- SCSI support
- SCSI generic support
- Support for the core module of your ppSCSI controller (t348 for the APA−348 and T348, t358 for the APA−358 and T358, epsa2 for the older Shuttle EPSA−2, epst for the Shuttle EPST and APA−1350, onscsi for the OnSpec 90c26, and sparcsi for the SparCSI and ParaSCSI)

Once these are compiled in, it's simply a matter of loading the appropriate modules.

3. Making and Accessing the Scanner Devices

The following section applies to all hardware types. Some specifics with regard to scanner interface types are mentioned in the paragraphs at the end of this section.

3.1. Device Filesystem and Udev

Devfs, or 'device filesystem' has been an option in the Linux kernel since the late 2.2−series. Devfsd, the device filesystem daemon, creates and removes devices on your system dynamically without the need to manually create devices. If you are running devfsd/devfs you can probably skip the following sections as the process of creating device nodes will be done for you and it's simply a matter of finding the appropriate device node in `/dev`.

Devfs does not obviate the need to change permissions of devices for access by users.

Beginning in the 2.6−series kernel devfs has been deprecated in favor of a userspace daemon known as udev, though devfs remains as an option. You can find information on udev here.

3.2. Creating Devices Manually

If you are running a system with correctly configured devfs, udev or libusb, you can skip this step and go to Section 4. There are two ways to accomplish the creation of necessary devices manually. One is to use **MAKEDEV** and the other is to create the device nodes at the command line.

The **MAKEDEV** script is the easier of the two methods, the executable of which may be located in `/dev` or the usual places for storing binary executables (`/bin`,`/sbin` and so on). I direct you to **man MAKEDEV**, and would caution you to pay attention to the device−specific command options so that you can be sure the major and minor numbers are correct (see the next paragraph for more on this and why it is important, especially if **MAKEDEV** doesn't work or you prefer doing things the hard way).

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.

3.2.1. SCSI Devices

If you are running a 2.4−series kernel you should consider becoming familiar with SCSI proc interface access, and whichever kernel you are running, you should read **man sane−scsi** before reading further. When the system boots up, generic SCSI device files are mapped on `/dev/sgN`, where N is a numeric value starting at zero. The major and minor numbers for SCSI devices are 21 and 0,1,2,3... respectively. You can find out what devices are loaded already with **ls −l /dev/sg***, which should yield output similar to this:

```
crw-------   1 root   sys     21,  0 Jan 06  2003 /dev/sg0
crw-------   1 root   sys     21,  0 Jan 06  2003 /dev/sg1
crw-------   1 root   sys     21,  0 Jan 06  2003 /dev/sg2
crw-------   1 root   sys     21,  0 Jan 06  2003 /dev/sg3
crw-------   1 root   sys     21,  0 Jan 06  2003 /dev/sg4
crw-------   1 root   sys     21,  0 Jan 06  2003 /dev/sg5
```

You will need to make a `/dev/scanner` symbolic link to an existing device (for reasons clarified later). For example, if your scanner is connected to the first scsi−bus (and lun and target) of your SCSI host device, you should link it to the corresponding device:

```
#  ln −s /dev/sg0 /dev/scanner
```

3.2.2. Manually creating USB Devices

Again, you can skip this step if using libusb. USB scanner devices have the major number 180 and minor 48, 49, etc., up to 63. First, check `/dev` to see what directory your distribution lays out its USB directory devices in, as some distributions might have these devices `scanner0, scanner1...`etc., within `/dev/usb` or as `usbscanner0, usbscanner1...` and so on, in the base `/dev/` directory. If you find that in the `/dev/` directory the scanner devices have already been made for you then your work is done. If not, you will need to create them yourself. As root, make a character device for your scanner like so:

```
# mknod /dev/usbscanner0 c 180 48
```

...or if your distribution has a '/dev/usb' subdirectory:

```
# mknod /dev/usb/scanner0 c 180 48
```

3.2.3. Manually creating Parallel Port Devices

Follow the example outlined in the above section to create the following generic parport devices:

```
crw−−−−−−−    1 root     root      99,   0 Jun 24 13:47 parport0
crw−−−−−−−    1 root     root      99,   1 Jun 24 13:47 parport1
crw−−−−−−−    1 root     root      99,   2 Jun 24 13:47 parport2
crw−−−−−−−    1 root     root      99,   3 Jun 24 13:47 parport3
crw−r−−−−−    1 root     root       1,   4 Jan  1  1970 port
```

You may also need to create `/dev/port` and/or `/dev/parport` depending on the backend you will use, so be prepared to return to this step if your application dictates it.

3.3. 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 scanner access to a particular group. On my system, the members of the group 'scanner' are allowed to use the scanner. The way to accomplish this is to first change the ownership of the devices in `/dev` like so (as root):

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

...where **root.scanner** 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, whether `/dev/sg*` on SCSI scanners, etc. 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 scanner /etc/group**. You should see something like the following:

```
scanner:x:103:
```

...where '103' is the group number. Since no members follow the last colon in the 'scanner' group we can add them, let's say user 'jhs' with the command

```
# adduser jhs scanner
```

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/usb/scanner0
```

...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. SANE

The final prerequisite for scanner access is the SANE backend(s) and optionally, a suitable SANE−frontend. The former are the drivers and low−level access tools that interface with your scanner, and the latter are graphical applications for access and use of your scanner within X. Only the former are required for scanner access, though a frontend is highly recommended in order to manipulate images and to actually be able to see your images in a windowed environment without having to print them.

4.1. Getting SANE

You can acquire the suite of SANE backends at http://www.sane−project.org/source.html, where you can obtain binaries for nearly all Linux distributions as well as source code. If you are planning on compiling from source, you probably already know what to do, but the following link is available for those that want a refresher, that of the Software Building HOWTO. In addition, be sure that if you have a previous sane installation that it is removed prior to installing your freshly−compiled version, and that you should acquire the most recently released stable version of the source code for compiling.

Those who wish to install binaries should download the corresponding file and then install as usual, i.e. for rpm−based distributions:

```
# rpm −iVh sane−backends−VERSION.rpm
```

For Debian users there is a SANE package in stable (Woody), testing (Sarge) and unstable (Sid) package repositories, so a simple **apt−get install sane** is all that is required, whatever version you are using.

Those who prefer compiling the latest version of SANE from source can acquire it from ftp.mostang.com/pub/sane. There is a more in−depth (though rather pessimistic) write−up of how to compile SANE from source and get a SCSI scanner working from scratch, at Laurent−jan's HOWTO page originally written by Steve Sheriff (the graphics are interesting, too).

4.2. Configuring SANE

4.2.1. SANE Backends

Whether you obtained your distribution's official SANE package, obtained a binary from the SANE homepage or compiled your own SANE binary from source, SANE should identify the appropriate backend to use for your hardware when you call **scanimage** or any other frontend. If no device is found when you run **scanimage −L** or your chosen frontend, see Section 7 for more info.

4.2.2. Across a Network

If you are interested in making scanner services available across a network from or to a remote machine, you will need to edit the `saned.conf` file in the configuration directory of the *server* (the computer with the scanner), whether `/etc/sane.d` or `/usr/local/etc/sane.d`. It usually consists of an entry 'scan−client.somedomain.firm' that will need to be replaced with the hostname of the client you want to be able to use the server's scanner. If you prefer an IP address this can be used instead.

The saned daemon will need to be run as well as inetd or xined on the server. See **man saned** for the exact changes required to inetd.conf or xined.conf. In addition port 6566 will need to be added to the `/etc/services` file:

```
sane 6566/tcp
```

The client computer (without the scanner) will need net.conf edited to include the server machine name, i.e., 'scan−server.somedomain.firm.'

Also for the client(s), be sure the entry "net" isn't commented out in the `dll.conf` file.

4.2.3. Using SANE with a Video4linux Device

Video4linux devices include webcams, still cameras and video capture devices. SANE is capable of accessing these. To do this, locate the file in the configuration directory (`/etc/sane.d` or `/usr/local/etc/sane.d`) named `v4l.conf`. Opening this file yields something similar to the following content:

```
# In order to use the v4linux backend you have to give the device
# You can enable multiple lines if
# you really have multible [sic] v4l devices.
#
/dev/bttv0
/dev/video0
/dev/video1
/dev/video2
/dev/video3
```

The initial line of this file really tells you all you need to know, so remember this when we get to the sections on testing the scanner hardware. Just be sure that whatever device your kernel identifies your camera or other v4l device as is uncommented (i.e., has the # removed from in front of it as above). You will obviously need to do this as root. In addition, be sure the line 'v4l' isn't commented out in the `dll.conf` file.

5. Testing Your Scanner

Once you've completed all of the above, you're ready to test your scanner equipment. This section assumes your scanner is turned on and has been attached through the appropriate interface. If you have a SCSI or a USB scanner, at the command line you can issue the following command:

```
$ sane−find−scanner
```

...which should find and identify your scanner from a list of possible devices. (Note to Debian users: starting with Sarge, or unstable, sane−find−scanner is available in the 'sane−utils' package. In Woody it is available in the 'libsane' package.) If your scanner is a type not looked for by **sane−find−scanner**, you can try as root **scanimage −−list−devices** which should yield information about attached devices. For example, this is the output on my system:

```
device `v4l:/dev/video0' is a Noname BT878 video (Hauppauge (bt878)) virtual device
device `epson:/dev/scanner0' is a Epson Perfection1240 flatbed scanner
```

whereas when using libusb it registers as

```
device `v4l:/dev/video0' is a Noname BT878 video (Hauppauge (bt878)) virtual device
device `epson:libusb:001:003' is a Epson Perfection1240 flatbed scanner
```

Make note of the 'backend:device' information obtained; this will be our device name to specifically access the scanner from the command line. Also, be aware that sane−find−scanner is a separate utility that does not guarantee support under SANE, it only looks for devices that claim to be scanners.

Next you should test the scanner's image grabbing ability. You can use either one of the frontends listed in Section 6 or at the command line if you wish with the following:

```
$ scanimage −d backend:/dev/scanner −−format pnm > outfile.pnm
```

Thus if you use the Epson backend, for example, the command would be as follows:

```
$ scanimage −d epson:/dev/scanner −−format pnm > outfile.pnm
```

You only need the **−d** option if you have more than one scanner and want to select which one to use. For example, if you have an Epson and a Mustek scanner, using "−d epson" or "−d mustek" should be enough. The complete path is only needed if you have more that one scanner supported by the same backend. Obviously **/dev/scanner** should be substituted with whatever scanner device you've configured (e.g., `/dev/video0` in the case of a v4l device, and libusb as seen in the sane−find−scanner example above). The **−−format** switch can be either pnm or tiff, but if left out will default to pnm. See **man scanimage** for more obscure but useful options. The '.pnm' format stands for 'portable anymap,' a common image format for graphical files in Linux that can be converted to nearly any other image format with Imagemagick or netpbm.

6. SANE Frontends

Now that you've got the hardware working, you should probably acquire a suitable frontend if you plan on using your scanner device in X11, which is probably a good idea to look at what you've scanned. My personal favorite is as elegant and functional as any proprietary solution I've seen, xsane. It has an attractive GTK+ based GUI, can save the image to a variety of formats, send an image to the printer, and interface easily with the GIMP. It makes accessing the full color and other potential of your hardware quite easy.

The GIMP, or GNU Image Manipulation Program, is an outstanding application for image editing if you are interested in scanning from within a Photoshop"−like application. The xsane module may be available as a separate package depending on your Linux distribution. After starting GIMP, click 'File,' then 'Acquire' followed by 'Xsane:device dialog' to access your scanner.

Another highly recommended frontend is Kooka of the KDE desktop environment. It has an intuitive interface that integrates easily with other KDE applications and can greatly simplify management of large image collections.

Xscanimage is a somewhat simpler (but still powerful) scanner application for X11 to acquire images from your scanner. It may or may not come bundled with the SANE backends depending on your distribution. See **man xscanimage** for more info.

You can find a more complete list of SANE frontends at the SANE frontends page.

7. Troubleshooting

7.1. Help, my scanner cannot be found by scanimage or xsane!

First, don't despair. If you're sure you've done everything correctly up to this point, all the right modules are loaded and all the configuration files tweaked as outlined and you know you're hardware is supported, check your permissions. In order to access scanner hardware you must have read and write access. See Section 3.3 for more info. If this isn't the problem, go to `/etc/sane.d/` (or `/usr/local/etc/sane.d`) and edit the file `dll.conf`, commenting out any backend or other (i.e. v4l) protocol that you don't need.

If none of the above work, from within the directory containing the SANE configuration files, open the one named after the backend for your particular scanner. There are (among others) two important entries in the file: interface type (scsi vs. usb), and the device name. If you have a usb scanner, you will usually need to comment out (make a # mark in front of) the 'scsi' line, and uncomment the line containing 'usb.' In addition the device name may need to be changed, depending on your distribution (i.e., `/dev/usbscanner0` may become `/dev/usb/usbscanner0`). As you may have noted, there may be several other options available to your scanner in this file depending on the model, so if your scanner doesn't operate as planned, you may want to take a look at this file and the accompanying model−specific documentation if any; see **man sane−scsi** or **sane−usb**, or whichever manufacturer made your scanner (including **sane−plustek**, **sane−qcam**, **sane−ricoh**, **sane−sharp**, **sane−snapscan**, **sane−umax** and so on. For a full list try **apropos sane**. The exact protocols and manufacturers available may depend on your version of SANE.

If none of the above work, see Section 7.5. Also, if you're particularly daring you should check the sane−troubleshoot Homepage, still in an early development stage at the time of this writing.

7.2. Help, I'm not sure my USB hardware is working!

Assuming you have usbdevfs and /proc filesystem support, you should issue the following command: **cat /proc/bus/usb/devices**. It should give you an output of the USB bus status and the connected devices and troubleshoot your hardware. If your scanner is supported and you can see your hardware you'll know your problem lies elsewhere.

7.3. Help, scanimage or the frontend I am using identifies the wrong device!

First, locate your configuration files, located in one of the usual places: `/etc/sane.d` or `/usr/local/etc/sane.d`. In general, if you obtained a precompiled package from your distribution or a binary from the SANE homepage it is in `/etc`, while if you compiled it from source it is in `/usr/local/etc/sane.d`. Change (**cd**) to that particular directory. In Section 2 you were referred to the SANE list of supported and not−yet−supported hardware. There you will find among the charts of individual manufacturers listed the "Backend," or SANE driver for each model in addition to the support status. Within `/etc/sane.d` or `/usr/local/etc/sane.d` there are similarly named files for each backend. You should select the file named `dll.conf`. This will list the backend protocols one by one. Check to be sure your scanner's backend is not commented out (i.e., has a hash mark in front of it). If it is you will need to (as root, and using your editor program) remove the '#.' If you still can't get things to work, see Section 7.5

7.4. Help, I can only access my parallel−port scanner as root!

The SANE driver for your scanner accesses the parallel port directly (via `/dev/port`). This only works for root for security reasons. See this mini−HOWTO by Till Kamppeter for instructions on how to approach this problem.

7.5. Help, I have an Acme Whizzbang" or other model scanner and you haven't addressed my particular problem!

Go to the mailing list and irc channel at the SANE website. Check the link for instructions on how to subscribe, etc. Also, you should read the SANE FAQ which has several hardware−specific questions, answers and links to relevant documentation.

---

8. Gnu Free Documentation License

Version 1.2, November 2002

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 whose 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 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 whose 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 5 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 whose 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 whose 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 whose 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.