NAME

sane−usb − USB configuration tips for SANE

DESCRIPTION

This manual page contains information on how to access scanners with a USB interface. It focusses on two main topics: getting the scanner detected by the operating system kernel and using it with SANE.

This page applies to USB most backends and scanners, as they use the generic sanei_usb interface. However, there is one exceptions: USB Scanners supported by the microtek2 backend need a special USB kernel driver, see **sane−microtek2**(5) for details.

QUICK START

This is a short HOWTO-like section. For the full details, read the following sections. The goal of this section is to get the scanner detected by **sane−find−scanner**(1).

Run sane−find−scanner. If it lists your scanner with the correct vendor and product ids, you are done. See section **SANE ISSUES** for details on how to go on.

sane−find−scanner doesn't list your scanner? Does it work as root? If yes, there is a permission issue. See the **LIBUSB** section for details.

Nothing is found even as root? Check that your kernel supports USB and that libusb is installed (see section **LIBUSB**).

USB ACCESS METHODS

For accessing USB devices, the USB library libusb is used. There used to exist another method to access USB devices: the kernel scanner driver. The kernel scanner driver method is deprecated and shouldn't be used anymore. It may be removed from SANE at any time. In Linux, the kernel scanner driver has been removed in the 2.6.* kernel series. Only libusb access is documented in this manual page.

LIBUSB

SANE can only use libusb 0.1.6 or newer. It needs to be installed at build-time. Modern Linux distributions and other operating systems come with libusb.

Libusb can only access your scanner if it's not claimed by the kernel scanner driver. If you want to use libusb, unload the kernel driver (e.g. rmmod scanner under Linux) or disable the driver when compiling a new kernel. For Linux, your kernel needs support for the USB filesystem (usbfs). For kernels older than 2.4.19, replace "usbfs" with "usbdevfs" because the name has changed. This filesystem must be mounted. That's done automatically at boot time, if /etc/fstab contains a line like this:

```
none /proc/bus/usb usbfs defaults  0  0
```

The permissions for the device files used by libusb must be adjusted for user access. Otherwise only root can use SANE devices. For *Linux*, the devices are located in /proc/bus/usb/ or in /dev/bus/usb, if you use udev. There are directories named e.g. "001" (the bus name) containing files "001", "002" etc. (the device files). The right device files can be found out by running scanimage −L as root. Setting permissions with "chmod" is not permanent, however. They will be reset after reboot or replugging the scanner.

Usually udev or for older distributions the hotplug utilities are used, which support dynamic setting of access permissions. SANE comes with udev and hotplug scripts in the directory tools/udev and tools/hotplug. They can be used for setting permissions, see @DOCDIR@/README.linux, tools/README and the README in the tools/hotplug directory for more details.

For the **BSDs**, the device files used by libusb are named */dev/ugen**. Use chmod to apply appropriate permissions.

SANE ISSUES

This section assumes that your scanner is detected by sane−find−scanner. It doesn't make sense to go on, if this is not the case. While sane−find−scanner is able to detect any USB scanner, actual scanning will only

work if the scanner is supported by a SANE backend. Information on the level of support can be found on the SANE webpage (*http://www.sane−project.org/*), and the individual backend manpages.

Most backends can detect USB scanners automatically using "usb" configuration file lines. This method allows one to identify scanners by the USB vendor and product numbers. The syntax for specifying a scanner this way is:

usb *VENDOR PRODUCT*

where *VENDOR* is the USB vendor id, and *PRODUCT* is the USB product id of the scanner. Both ids are non-negative integer numbers in decimal or hexadecimal format. The correct values for these fields can be found by running sane−find−scanner, looking into the syslog (e.g., /var/log/messages) or under Linux by issuing the command "cat /proc/bus/usb/devices". This is an example of a config file line:

usb 0x055f 0x0006

would have the effect that all USB devices in the system with a vendor id of 0x55f and a product id of 0x0006 would be probed and recognized by the backend.

If your scanner is not detected automatically, it may be necessary to edit the appropriate backend configuration file before using SANE for the first time. For a detailed description of each backend's configuration file, please refer to the relevant backend manual page (e.g. **sane−mustek_usb**(5) for Mustek USB scanners).

Do **not** create a symlink from */dev/scanner* to the USB device because this link is used by the SCSI backends. The scanner may be confused if it receives SCSI commands.

ENVIRONMENT

SANE_DEBUG_SANEI_USB

If the library was compiled with debug support enabled, this environment variable controls the debug level for the USB I/O subsystem. E.g., a value of 128 requests all debug output to be printed. Smaller levels reduce verbosity. Values greater than 4 enable libusb debugging (if available). Example: export SANE_DEBUG_SANEI_USB=4.

SANE_USB_WORKAROUND

If your scanner does not work when plugged into a USB3 port, try setting the environment variable SANE_USB_WORKAROUND to 1. This may work around issues which happen with particular kernel versions. Example: export SANE_USB_WORKAROUND=1.

SEE ALSO

sane(7), **sane−find−scanner**(1), **sane−"backendname"**(5), **sane−scsi**(5)

AUTHOR

Henning Meier-Geinitz <henning@meier−geinitz.de>