

NAME

sane-scsi – SCSI adapter tips for scanners

DESCRIPTION

This manual page contains various operating-system specific tips and tricks on how to get scanners with a SCSI interface working.

GENERAL INFO

For scanners with a SCSI interface, it may be necessary to edit the appropriate backend configuration file before using SANE for the first time. For most systems, the configuration file should list the name of the generic SCSI device that the scanner is connected to (e.g., under Linux, `/dev/sg4` or `/dev/sge` is such a generic SCSI device). It is customary to create a symlink from `/dev/scanner` to the generic SCSI device that the scanner is connected to. In this case, the configuration file simply lists the line `/dev/scanner`. For a detailed description of each backend's configuration file, please refer to the relevant backend manual page (e.g., `sane-epson(5)` for Epson scanners, `sane-hp(5)` for HP scanners, etc.).

For some operating systems (e.g. Linux and OS/2), there is an alternate way of specifying scanner devices. This alternate way allows one to identify scanners by the SCSI vendor and model string and/or by the SCSI device address (consisting of bus number, channel number, id, and logical unit number). The syntax for specifying a scanner in this way is:

```
scsi VENDOR MODEL TYPE BUS CHANNEL ID LUN
```

where *VENDOR* is the SCSI vendor string, *MODEL* is the SCSI model string, *TYPE* is type SCSI device type string, *BUS* is the SCSI bus number (named "host" in `/proc/scsi/scsi`), *CHANNEL* is the SCSI channel number, *ID* is the SCSI id, and *LUN* is the logical unit number of the scanner device. The first two fields are strings which must be enclosed in double-quotes if they contain any whitespace. The remaining four fields are non-negative integer numbers. The correct values for these fields can be found by using operating system specific tools, e.g. for Linux by looking at the output of the command "cat `/proc/scsi/scsi`". To simplify configuration, a field's value can be replaced with an asterisk symbol ("*"). An asterisk has the effect that any value is allowed for that particular field. This can have the effect that a single scsi-line matches multiple devices. When this happens, each matching device will be probed by the backend one by one and registered if the backend thinks it is a compatible device. For example, the line

```
scsi MUSTEK MFS-06000CX Scanner 0 00 03 00
```

would attach the Mustek SCSI scanner with the following `/proc/scsi/scsi` entry:

```
Host: scsi0 Channel: 00 Id: 03 Lun: 00
Vendor: MUSTEK    Model: MFS-06000CX Rev: 4.04
Type:   Scanner  ANSI SCSI revision: 0
```

Usually it's sufficient to use vendor and model strings only or even only the vendor string. The following example

```
scsi MUSTEK * * * * *
```

would have the effect that all SCSI devices in the system with a vendor string of MUSTEK would be probed and recognized by the backend.

If the remainder of a scsi-string consists of asterisks only, the asterisks can be omitted. For example, the following line is equivalent to the one specified previously:

```
scsi MUSTEK
```

On some platforms (e.g., OpenStep), SANE device names take a special form. This is explained below in the relevant platform-specific section.

When using a SCSI scanner, ensure that the access permission for the generic SCSI device is set appropriately. We recommend to add a group "scanner" to `/etc/group` which contains all users that should have access to the scanner. The permission of the device should then be set to allow group read and write access. For example, if the scanner is at generic SCSI device `/dev/sg0`, then the following two commands would set the permission correctly:

```
$ chgrp scanner /dev/sg0
$ chmod 660 /dev/sg0
```

When your system uses the device filesystem (devfs), you have to edit **/etc/devfs/perms**. There you should search the line

```
REGISTER ^sg[^\/*] PERMISSIONS root.root 0600
```

and add a new line (eg. for changing permissions of sg4):

```
REGISTER ^sg4 PERMISSIONS root.scanner 0660
```

FREEBSD INFO

Auto-configuration using the "scsi *" lines in the config files only works if the user running the frontend has read/write access to /dev/xpt0. Instead, you can also set a link */dev/scanner* to the appropriate /dev/uk device.

Adaptec AHA1542CF

Reported to work fine under FreeBSD 2.2.2R with the **aha** driver.

Adaptec 2940

Reported to work fine under FreeBSD 2.2.2.

Adaptec 1522

The scanner probes ok but any attempt to access it *hangs* the entire system. It looks like something is disabling interrupts and then not re-enabling them, so it looks like a bug in the FreeBSD **aic** driver.

Adaptec 1505

Works on FreeBSD 2.2.5R and 3.0 using the **aic** driver, provided that Plug-and-Play support is disabled on the card. If there are no *uk* devices, just do a "sh MAKEDEV uk0" in the **/dev** directory. The scanner should then be accessible as **/dev/uk0** if it was probed during boot.

Tekram DC390

Reported to work fine under FreeBSD 2.2.2R with the **amd** driver.

LINUX INFO

First, make sure your kernel has SCSI generic support enabled. In "make xconfig", this shows up under "SCSI support->SCSI generic support".

To keep scanning times to a minimum, it is strongly recommended to use a large buffer size for the generic SCSI driver. From SG driver version 2.0 on, the maximum buffer size can be changed at program run time, and there is no restriction in size. This driver version is part of the Linux kernels from version 2.2.7 on. If the new SG driver is available some backends (e.g. sane-umax, sane-mustek, sane-sharp) automatically request larger scsi buffers. If a backend does not automatically request a larger scsi buffer, set the environment variable **SANE_SG_BUFFERSIZE** to the desired buffer size in bytes. It is not recommended to use more than 1 MB, because for large values the probability increases that the SG driver cannot allocate the necessary buffer(s). For ISA cards, even 1 MB might be a too large value. For a detailed discussion of memory issues of the SG driver, see <http://www.torque.net/sg>.

For Linux kernels before version 2.2.7 the size of the buffer is only 32KB. This works, but for many cheaper scanners this causes scanning to be slower by about a factor of four than when using a size of 127KB. Linux defines the size of this buffer by macro **SG_BIG_BUFF** in header file */usr/include/scsi/sg.h*. Unless a system is seriously short on memory, it is recommended to increase this value to the maximum legal value of 128*1024-512=130560 bytes. After changing this value, it is necessary to re-compile both the kernel (or the SCSI generic module) and the SCSI backends. Keep in mind that this is only necessary with older Linux kernels.

A common issue with SCSI scanners is what to do when you booted the system while the scanner was turned off? In such a case, the scanner won't be recognized by the kernel and SANE won't be able to

access it. Fortunately, Linux provides a simple mechanism to probe a SCSI device on demand. Suppose you have a scanner connected to SCSI bus 2 and the scanner has a SCSI id of 5. When the system is up and running and the scanner is turned on, you can issue the command:

```
echo "scsi add-single-device 2 0 5 0" > /proc/scsi/scsi
```

and the kernel will probe and recognize your scanner (this needs to be done as root). It's also possible to dynamically remove a SCSI device by using the "remove-single-device" command. For details, please refer to the SCSI-2.4-HOWTO.

Scanners are known to work with the following SCSI adapters under Linux. This list isn't complete, usually any SCSI adapter supported by Linux should work.

Acard/Advance SCSI adapters

Some old versions of the kernel driver (atp870u.c) cut the inquiry information. Therefore the scanner couldn't be detected correctly. Use a current kernel.

Adaptec AHA-1505/AHA-1542/AHA-2940

Reported to work fine with Linux since v2.0. If you encounter kernel freezes or other unexpected behaviour get the latest Linux kernel (2.2.17 seems to work) or reduce SCSI buffer size to 32 kB.

ASUS SC200

Reported to work fine with Linux v2.0.

BusLogic BT958

To configure the BusLogic card, you may need to follow these instructions (contributed by Jeremy <jeremy@xxedgexx.com>): During boot, when your BusLogic adapter is being initialized, press Ctrl-B to enter your BusLogic adapter setup. Choose the address which your BusLogic containing your scanner is located. Choose "SCSI Device Configuration". Choose "Scan SCSI Bus". Choose whatever SCSI id that contains your scanner and then choose "View/Modify SCSI configuration". Change "Negotiation" to "async" and change "Disconnect" to "off". Press Esc, save, and Esc again until you are asked to reboot.

NCR/Symbios 53c400/53c400a or Domex DTC3181E/L/LE (DTCT436/436P) ISA SCSI card

This card is supplied by Mustek (and other vendors). It's supported since Linux 2.2. The SCSI cards are supported by the module g_NCR5380. It's necessary to tell the kernel the io port and type of card. Example for a 53c400a: "modprobe g_NCR5380 ncr_addr=0x280 ncr_53c400a=1". Once the kernel detects the card, it should work all right. However, while it should work, do not expect good performance out of this card---it has no interrupt line and therefore while a scan is in progress, the system becomes almost unusable. You may change the values of the USLEEP macros in drivers/scsi/g_NCR5380.c. Some documentation is in this file and NCR5380.c.

NCR/Symbios 810

For some scanners it may be necessary to disable disconnect/reconnect. To achieve this use the option ncr53c8xx="disc:n". Some people reported that their scanner only worked with the 53c7,8xx driver, not the ncr53c8xx. Try both if you have trouble.

For Linux kernels before 2.0.33 it may be necessary to increase the SCSI timeout. The default timeout for the Linux kernels before 2.0.33 is 10 seconds, which is way too low when scanning large area. If you get messages of the form "restart (ncr dead ?)" in your /var/log/messages file or on the system console, it's an indication that the timeout is too short. In this case, find the line "if (np->latetime>10)" in file ncr53c8xx.c (normally in directory /usr/src/linux/drivers/scsi) and change the constant 10 to, say, 60 (one minute). Then rebuild the kernel/module and try again.

Tekram DC315

The driver can be downloaded from <http://www.garloff.de/kurt/linux/dc395/>. For some older scanners it may be necessary to disable all the more advanced features by using e.g.

```
modprobe dc395x_trm dc395x_trm=7,5,1,32.
```

Tekram DC390

Version 1.11 of the Tekram driver seems to work fine mostly, except that the scan does not terminate properly (it causes a SCSI timeout after 10 minutes). The generic AM53C974 also seems to work fine and does not suffer from the timeout problems.

SOLARIS, OPENSTEP AND NEXTSTEP INFO

Under Solaris, OpenStep and NeXTStep, the generic SCSI device name refers to a SCSI bus, not to an individual device. For example, **/dev/sg0** refers to the first SCSI bus. To tell SANE which device to use, append the character 'a'+target-id to the special device name. For example, the SCSI device connected to the first SCSI controller and with target-id 0 would be called **/dev/sg0a**, and the device with target-id 1 on that same bus would be called **/dev/sg0b**, and so on.

ENVIRONMENT

SANE_DEBUG_SANEI SCSI

If the library was compiled with debug support enabled, this environment variable controls the debug level for the generic SCSI I/O subsystem. E.g., a value of 128 requests all debug output to be printed by the backend. A value of 255 also prints kernel messages from the SCSI subsystem (where available). Smaller levels reduce verbosity.

SANE_SCSCMD_TIMEOUT

sets the timeout value for SCSI commands in seconds. Overriding the default value of 120 seconds should only be necessary for very slow scanners.

SEE ALSO

sane(7), **sane-find-scanner(1)**, **sane-"backendname"(5)**, **sane-usb(5)**

AUTHOR

David Mosberger