



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'sane-sharp.5'

\$ man sane-sharp.5

sane-sharp(5) SANE Scanner Access Now Easy sane-sharp(5)

NAME

sane-sharp - SANE backend for SHARP scanners

DESCRIPTION

The sane-sharp library implements a SANE (Scanner Access Now Easy) backend that provides access to Sharp SCSI scanners. This backend should be considered beta-quality software!

In the current state it is known to work with JX-610 and JX-250 scanners. It is prepared for usage with the JX-330 series scanners, but we are not able to test it with these devices.

For other Sharp scanners, it may or may not work.

At present, the following scanners are known to work with this backend.

Vendor Product id:

Sharp JX-610

Sharp JX-250

Sharp JX-320

Sharp JX-330

Sharp JX-350

The following scanners are detected by the backend, but not tested:

Vendor Product id:

Sharp JX-325

DEVICE NAMES

This backend expects device names of the form:

special

where special is the path-name for the special device that corresponds to a SCSI scanner.

The special device name must be a generic SCSI device or a symlink to such a device. Under Linux, such a device name could be /dev/sga or /dev/sge, for example. See sane-scsi(5) for details.

SCAN OPTIONS

--mode Scan Mode. Possible settings are: Linear (1 bit black & white scans), Gray (8 bit gray scale scans), Linear Color (bi-level color scans), and Color (8 bit RGB scans). The default value is Color.

--halftone-pattern

Halftone Pattern. Available only for the JX-330 series scanners. Possible settings: none, Dither Bayer, Dither Spiral, Dither Dispersed and Error Diffusion. The default value is none.

--source

Paper Source. This option is only available if an automatic document feeder or a transparency adapter is installed. Possible settings are: Flatbed, Automatic Document Feeder, and Transparency Adapter. If an ADF or a transparency adapter is installed, using it is the default selection.

--custom-gamma

Custom Gamma This option determines whether a builtin or a custom gamma table is used. Possible settings are: yes (enables custom gamma tables) or no (enables a built gamma table).

--gamma

Gamma. This option is only available if Custom Gamma is set to no. Possible values are: 1.0 or 2.2. The default value is 2.2. (The JX-250 and JX-350 have no builtin gamma correction; for these scanners, a gamma table is downloaded to the scanner by the backend.)

--gamma-table

Gamma Table. Allowed values: 0..255; 256 numbers must be defined. The default values are 0, 1, 2, .. 255 (i.e., gamma == 1). This table is only used for gray scale scans.

--red-gamma-table

Red Gamma Table. Allowed values: 0..255; 256 numbers must be defined. The default values are 0, 1, 2, .. 255 (i.e., gamma == 1).

--green-gamma-table

Green Gamma Table. Allowed values: 0..255; 256 numbers must be defined. The default values are 0, 1, 2, .. 255 (i.e., gamma == 1).

--blue-gamma-table

Blue Gamma Table. Allowed values: 0..255; 256 numbers must be defined. The default values are 0, 1, 2, .. 255 (i.e., gamma == 1).

--resolution

Selects the resolution of the scanned image. Allowed values: 30..600 (JX-330, JX-350 and JX-610) and 30..400 (JX-250). The default value is 150.

-l, -t, -x, -y

Scan Window. Top-left x position of scan area (-l), top-left y position of scan area (-t), bottom right x position of scan area (-x) and bottom right y position of scan area (-y). The possible settings depend on the scanner model and, for the JX-250 and the JX-350, also on the usage of the automatic document feeder resp. the transparency adapter. Please refer to the values allowed by xscanimage(1), or xsane(1). With scanimage(1), enter one of the following commands in order to see the allowed parameter values for the scan window:

```
scanimage -d sharp --source "Automatic Document Feeder" --help
```

```
scanimage -d sharp --source Flatbed --help
```

```
scanimage -d sharp --source "Transparency Adapter" --help
```

--edge emphasis

Edge emphasis. This option is not available for the JX-250 and the JX-350. Possible settings: None, Middle, Strong, and Blur. The default value is None.

--threshold

Sets the threshold for black and white pixels in lineart mode. Possible values are 1..255. The default value is 128. This option is only available in scan mode lineart.

--threshold-red

Sets the threshold for the red component of a pixel in in lineart color scan mode. Possible values are 1..255. The default value is 128. This option is only available in scan mode color lineart.

--threshold-green

Sets the threshold for the green component of a pixel in in lineart color scan mode. Possible values are 1..255. The default value is 128. This option is only available in scan mode color lineart .

--threshold-blue

Sets the threshold for the blue component of a pixel in in lineart color scan mode. Possible values are 1..255. The default value is 128. This option is only available in scan mode color lineart.

--lightcolor

Sets the color of the light source. Possible values are white, red, green and blue. The default value is white. This option is only available in scan modes lineart color and color.

ADF USAGE

If a paper jam occurs, the maintenance cover must be opened and closed, even if the jammed paper can be removed without opening the maintenance cover. Otherwise, the error condition will not be cleared.

CONFIGURATION

The contents of the sharp.conf file is a list of options and device names that correspond to Sharp scanners. Empty lines and lines beginning with a hash mark (#) are ignored. See sane-scsi(5) for details about device names.

Lines setting an option start with the key word option, followed by the option's name and the option's value. At present, three options are defined: buffers, buffersize, and read? queue.

Options defined at the start of sharp.conf apply to all devices; options defined after a device name apply to this device.

The options buffers and readqueue are only significant if the backend has been compiled so that for each scan a second process is forked (switch USE_FORK in sharp.c). This process reads the scan data from the scanner and writes this data into a block of shared memory.

The parent process reads the data from this memory block and delivers it to the frontend.

The options control the size and usage of this shared memory block.

option buffers defines the number of buffers used. The smallest number allowed is 2.

option buffersize defines the size of one buffer. Since each buffer is filled with a sin?

gle read command sent to the scanner, its size is limited automatically to the size al?

lowed by the operating system or by the Sane SCSI library for SCSI read commands. A buffer size of 128 kB or 256 kB is recommended for scan resolutions of 300 dpi and above.

option `readqueue` defines how many read commands to be sent to the scanner are queued. At present, the Sane SCSI library supports queued read commands only for Linux. For other operating systems, option `readqueue` should be set to 0. For Linux, option `readqueue` should be set to 2. Larger values than 2 for option `readqueue` are not reasonable in most cases. option `buffers` should be greater than option `readqueue`.

Performance Considerations

This section focuses on the problem of stops of the scanner's carriage during a scan. Carriage stops happen mainly with the JX-250. This scanner has obviously only a small internal buffer compared to its speed. That means that the backend must read the data as fast as possible from the scanner in order to avoid carriage stops.

Even the JX-250 needs only less than 10 seconds for a 400 dpi A4 gray scale scan, which results in a data transfer rate of more than 1.6 MB per second. This means that the data produced by the scanner must be processed fairly fast. Due to the small internal buffer of the JX-250, the backend must issue a read request for the next data block as soon as possible after reading a block of data in order to avoid carriage stops.

Stops of the carriage can be caused by the following reasons:

- too much "traffic" on the SCSI bus
- slow responses by the backend to the scanner,
- a program which processes the data acquired by the backend too slow.

Too much "traffic" on the SCSI bus: This happens for example, if hard disks are connected to the same SCSI bus as the scanner, and when data transfer from/to these hard disks requires a considerable part of the SCSI bandwidth during a scan. If this is the case, you should consider to connect the scanner to a separate SCSI adapter.

Slow responses by the backend to the scanner: Unfortunately, UNIX-like operating systems generally have no real time capabilities. Thus there is no guarantee that the backend is under any circumstances able to communicate with the scanner as fast as required. To minimize this problem, the backend should be compiled so that a separate reader process is forked: Make sure that `USE_FORK` is defined when you compile `sharp.c`. If slow responses of the backend remain to be problem, you could try to reduce the load of the system. Even while the backend and the reader process need only a minor amount of processor time, other running processes can cause an increase in the time delay between two time slices given to

the reader process. On slower systems, such an increased delay can be enough to cause a carriage stop with the JX-250. For Linux, the usage of the SG driver version 2.1.36 or above is recommended, because it supports, in combination with the SCSI library of Sane version 1.0.2, command queueing within the kernel. This queueing implementation, combined with a buffer size of at least 128 kB, should avoid most carriage stops.

Slow processing of the scan data: An example for this situation is the access to the scanner via a 10 MBit Ethernet, which is definitely too slow to transfer the scan data as fast as they are produced by the scanner. If you have enough memory available, you can increase option buffers, so that an entire image can be stored in these buffers.

In order to see, if the backend is too slow or if the further processing of the data is too slow, set the environment variable `SANE_DEBUG_SHARP` to 1. When a scan is finished, the backend writes the line "buffer full conditions: nn" to stderr. If nn is zero, carriage stops are caused by too slow responses of the backend or too much "traffic" on the SCSI bus. If nn is greater than zero, the backend had to wait nn times until a buffer has been processed by the frontend. (Please note that option buffers must be greater than option `readqueue` in order to get useful output for "buffer full conditions".)

FILES

`/etc/sane.d/sharp.conf`

The backend configuration file.

`/usr/lib/x86_64-linux-gnu/sane/libsane-sharp.a`

The static library implementing this backend.

`/usr/lib/x86_64-linux-gnu/sane/libsane-sharp.so`

The shared library implementing this backend (present on systems that support dynamic loading).

ENVIRONMENT

`SANE_DEBUG_SHARP`

If the library was compiled with debug support enabled, this environment variable controls the debug level for this backend. E.g., a value of 128 requests all debug output to be printed. Smaller levels reduce verbosity.

KNOWN PROBLEMS

1. ADF Mode

After several ADF scans, the scanner moves the carriage back to the idle position and back to ADF scan position, before a scan starts. We do not know, if this is a

problem of the scanner, or if this is a bug of the backend. At present, the scanner must power off and on to stop this annoying behaviour.

2. Threshold level does not work (only JX-610)

3. The maximum resolution is limited to 600 dpi(JX-610 supported to 1200 dpi) resp. 400 dpi (JX-250)

4. If the JX250 is used with an ADF, the following situation can occur: After several scans, the scanner moves, after loading a new sheet of paper, the carriage to the idle position, and then back to the position used for ADF scans. This happens for every scan, in contrast to the calibration, which is done after 10 scans. (For the calibration, the carriage is also moved to the idle position.) We do not know if this behavior is caused by the backend, or if it is a bug in the firmware of the scanner.

5. Usage of a transparency adapter (film scan unit) is supported, but not tested.

SEE ALSO

sane(7), sane-scsi(5)

AUTHORS

Kazuya Fukuda, Abel Deuring

CREDITS

The Sharp backend is based on the Canon backend written by Helmut Koeberle

Parts of this man page are a plain copy of sane-mustek(5) by David Mosberger-Tang, Andreas

Czechanowski and Andreas Bolsch

11 Jul 2008

sane-sharp(5)