



**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-bh.5'***

**\$ man sane-bh.5**

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

NAME

sane-bh - SANE backend for Bell+Howell Copiscan II series document scanners

DESCRIPTION

The sane-bh library implements a SANE (Scanner Access Now Easy) backend that provides access to Bell+Howell Copiscan II series document scanners. The Copiscan II 6338 has been the primary scanner model used during development and testing, but since the programming interface for the entire series is consistent the backend should work for the following scanner models:

- COPISCAN II 6338 Duplex Scanner with ACE
- COPISCAN II 2135 Simplex Scanner
- COPISCAN II 2137(A) Simplex Scanner (with ACE)
- COPISCAN II 2138A Simplex Scanner with ACE
- COPISCAN II 3238 Simplex Scanner
- COPISCAN II 3338(A) Simplex Scanner (with ACE)

If you have a Bell+Howell scanner and are able to test it with this backend, please contact [sane-devel@alioth-lists.debian.net](mailto:sane-devel@alioth-lists.debian.net) with the model number and testing results. Have a look at <http://www.sane-project.org/mailling-lists.html> concerning subscription to sane-devel. Additionally, the author is curious as to the likelihood of using this backend with the newer 4000 and 8000 series scanners. If you have such a beast, please let me know. The Bell+Howell Copiscan II series document scanners are high volume, high throughput scanners designed for document scanning applications. As such, they are lineart/grayscale scanners supporting a fixed number of fairly low resolutions (e.g. 200/240/300dpi). How?

ever, they do have a number of interesting and useful features suited to needs of document imaging applications. This backend attempts to support as many of these features as possible.

The main technical reference used in writing this backend is the Bell and Howell Copiscan II Remote SCSI Controller (RSC) OEM Technical Manual Version 1.5. The Linux SCSI programming HOWTO, the SANE API documentation, and SANE source code were also extremely valuable resources.

The latest backend release, additional information and helpful hints are available from the backend homepage:

<http://www.martoneconsulting.com/sane-bh.html>

## 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.

For SCSI scanners, the special device name must be a generic SCSI device or a symlink to such a device. Under Linux, such a device name takes a format such as /dev/sga or /dev/sg0, for example. See sane-scsi(5) for details.

## OPTIONS

Scan Mode Options:

--preview[=(yes|no)] [no]

Request a preview-quality scan. When preview is set to yes image compression is disabled and the image is delivered in a SANE\_FRAME\_GRAY frame.

--mode lineart|halftone [lineart]

Selects the scan mode (e.g., lineart,monochrome, or color).

--resolution 200|240|300dpi [200]

Sets the resolution of the scanned image. Each scanner model supports a list of standard resolutions; only these resolutions can be used.

--compression none|g31d|g32d|g42d [none]

Sets the compression mode of the scanner. Determines the type of data returned from the scanner. Values are:

none - uncompressed data - delivered in a SANE\_FRAME\_GRAY frame

g31d - CCITT G3 1 dimension (MH) - delivered in a SANE\_FRAME\_G31D frame

g32d - CCITT G3 2 dimensions (MR, K=4) - delivered in a SANE\_FRAME\_G32D frame

g42d - CCITT G4 (MMR) - delivered in a SANE\_FRAME\_G42D frame

NOTE: The use of g31d, g32d, and g42d compression values causes the backend to generate optional frame formats which may not be supported by all SANE frontends.

#### Geometry Options:

--autoborder[=(yes|no)] [yes]

Enable/Disable automatic image border detection. When enabled, the RSC unit automatically detects the image area and sets the window geometry to match.

--paper-size Custom|Letter|Legal|A3|A4|A5|A6|B4|B5 [Custom]

Specify the scan window geometry by specifying the paper size of the documents to be scanned.

--tl-x 0..297.18mm [0]

Top-left x position of scan area.

--tl-y 0..431.8mm [0]

Top-left y position of scan area.

--br-x 0..297.18mm [297.18]

Bottom-right x position of scan area.

--br-y 0..431.8mm [431.8]

Bottom-right y position of scan area.

#### Feeder Options:

--source Automatic Document Feeder|Manual Feed Tray [Automatic Document Feeder]

Selects the scan source (such as a document feeder). This option is provided to allow multiple image scans with xsane(1); it has no other purpose.

--batch[=(yes|no)] [no]

Enable/disable batch mode scanning. Batch mode allows scanning at maximum throughput by buffering within the RSC unit. This option is recommended when performing multiple pages scans until the feeder is emptied.

--duplex[=(yes|no)] [no]

Enable duplex (dual-sided) scanning. The scanner takes an image of each side of the document during a single pass through the scanner. The front page is delivered followed by the back page. Most options, such as compression, affect both the front and back pages.

--timeout-adf 0..255 [0]

Sets the timeout in seconds for the automatic document feeder (ADF). The value 0

specifies the hardware default value which varies based on the scanner model.

--timeout-manual 0..255 [0]

Sets the timeout in seconds for semi-automatic feeder. The value 0 specifies the hardware default value which varies based on the scanner model.

--check-adf[=(yes|no)] [no]

Check ADF Status prior to starting scan using the OBJECT POSITION command. Note that this feature requires RSC firmware level 1.5 or higher and dip switch 4 must be in the on position. NOTE: This option has not been tested extensively and may produce undesirable results.

Enhancement:

--control-panel[=(yes|no)] [yes]

Enables the scanner's control panel for selecting image enhancement parameters. When the option is set to no the following options are used to control image enhancement. See the Bell+Howell scanner users' guide for complete information on ACE functionality.

--ace-function -4..4 [3]

Specify the Automatic Contrast Enhancement (ACE) Function.

--ace-sensitivity 0..9 [5]

Specify the Automatic Contrast Enhancement (ACE) Sensitivity.

--brightness 0..255 [0]

Controls the brightness of the acquired image. Ignored for ACE capable scanners.

--threshold 0..255 [0]

Select minimum-brightness to get a white point. Ignored for ACE capable scanners.

--contrast 0..255 [inactive]

Controls the contrast of the acquired image. This option is not currently used by the scanner (and perhaps never will be).

--negative[=(yes|no)] [no]

Swap black and white, yielding a reverse-video image.

Icon:

--icon-width 0..3600pel (in steps of 8) [0]

Width of icon (thumbnail) image in pixels.

--icon-length 0..3600pel (in steps of 8) [0]

Length of icon (thumbnail) image in pixels.

## Barcode Options:

`--barcode-search-bar <see list> [none]`

Specifies the barcode type to search for. If this option is not specified, or specified with a value of none, then the barcode decoding feature is completely disabled. The valid barcode type are:

none

ean-8

ean-13

reserved-ean-add

code39

code2-5-interleaved

code2-5-3lines-matrix

code2-5-3lines-datalogic

code2-5-5lines-industrial

patchcode

codabar

codabar-with-start-stop

code39ascii

code128

code2-5-5lines-iata

`--barcode-search-count 1..7 [3]`

Number of times that the RSC performs the decoding algorithm. Specify the smallest number possible to increase performance. If you are having trouble recognizing barcodes, it is suggested that you increase this option to its maximum value (7).

`--barcode-search-mode <see list> [horiz-vert]`

Chooses the orientation of barcodes to be searched. The valid orientations are:

horiz-vert

horizontal

vertical

vert-horiz

`--barcode-hmin 0..1660mm [5]`

Sets the barcode minimum height in millimeters (larger values increase recognition speed). Of course the actual barcodes in the document must be of sufficient size.

--barcode-search-timeout 20..65535us [10000]

Sets the timeout for barcode searching in milliseconds. When the timeout expires, the decoder will stop trying to decode barcodes.

--section <string> []

Specifies a series of image sections. A section can be used to gather a subset image or to provide a small area for barcode decoding. Each section is specified in the following format (units are in millimeters):

<width>x<height>+<top-left-x>+<top-left-y>[:functioncode...]

Multiple sections can be specified by separating them with commas.

For example 76.2x25.4+50.8+0:frontbar identifies an area 3 inches wide and 1 inch high with a top left corner at the top of the page two inches from the left hand edge of the page. This section will be used for barcode decoding on the front page only.

For example 50.8x25.4+25.4+0:frontbar:front:g42d identifies an area 2 inches wide and 1 inch high with a top left corner at the top of the page one inch from the left hand edge of the page. This section will be used for barcode decoding on the front page as well as generating an image compressed in g42d format.

Ordinarily barcodes are searched in the entire image. However, when you specify sections all barcode searching is done within the specific sections identified. This can significantly speed up the decoding process.

The following function codes are available:

- front - generate an image for the front page section
- back - generate an image for the back page section
- frontbar - perform barcode search in front page section
- backbar - perform barcode search in back page section
- frontpatch - perform patchcode search in front page section
- backpatch - perform patchcode search in back page section
- none - use no image compression
- g31d - use Group 3 1 dimension image compression
- g32d - use Group 3 2 dimensions image compression
- g42d - use Group 4 2 dimensions image compression

If you omit a compression functioncode, the full page compression setting is used. If you specify multiple compression functioncodes, only the last one is used.

--barcode-relmax 0..255 [0]

Specifies the maximum relation from the widest to the smallest bar.

--barcode-barmin 0..255 [0]

Specifies the minimum number of bars in Bar/Patch code.

--barcode-barmax 0..255 [0]

Specifies the maximum number of bars in a Bar/Patch code.

--barcode-contrast 0..6 [3]

Specifies the image contrast used in decoding. Use higher values when there are more white pixels in the code.

--barcode-patchmode 0..1 [0]

Controls Patch Code detection.

## CONFIGURATION

The contents of the bh.conf file is a list of device names that correspond to Bell+Howell scanners. See sane-scsi(5) on details of what constitutes a valid device name. Additionally, options can be specified; these lines begin with the word "option". Each option is described in detail below. Empty lines and lines starting with a hash mark (#) are ignored.

## OPTIONS

The following options can be specified in the bh.conf file.

disable-optional-frames

This option prevents the backend from sending any optional frames. This option may be useful when dealing with frontends which do not support these optional frames.

When this option is in effect, the data is sent in a SANE\_FRAME\_GRAY frame. The optional frames sent by this backend are: SANE\_FRAME\_G31D, SANE\_FRAME\_G32D, SANE\_FRAME\_G42D and SANE\_FRAME\_TEXT. These frames are generated based on the compression and barcode options. These frames are never sent in preview mode.

fake-inquiry

This option is used for debugging purposes and its use is not encouraged. Essentially, it allows the backend to initialize in the absence of a scanner. This is useful for development and not much else. This option must be specified earlier in the configuration file than the devices which are to be "faked".

## FILES

/etc/sane.d/bh.conf

The backend configuration file (see also description of SANE\_CONFIG\_DIR below).

/usr/lib/x86\_64-linux-gnu/sane/libsane-bh.a

The static library implementing this backend.

/usr/lib/x86\_64-linux-gnu/sane/libsane-bh.so

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

## ENVIRONMENT

### SANE\_CONFIG\_DIR

This environment variable specifies the list of directories that may contain the configuration file. On \*NIX systems, the directories are separated by a colon (:), under OS/2, they are separated by a semi-colon (;). If this variable is not set, the configuration file is searched in two default directories: first, the current working directory (".") and then in /etc/sane.d. If the value of the environment variable ends with the directory separator character, then the default directories are searched after the explicitly specified directories. For example, setting SANE\_CONFIG\_DIR to "/tmp/config:" would result in directories tmp/config, ., and /etc/sane.d being searched (in this order).

### SANE\_DEBUG\_BH

If the library was compiled with debug support enabled, this environment variable controls the debug level for this backend. E.g., a value of 255 requests all debug output to be printed. Smaller levels reduce verbosity.

## SUPPORTED FEATURES

### ADF support

With document scanners, automatic document feeder (ADF) support is a key feature. The backend supports the ADF by default and returns SANE\_STATUS\_NO\_DOCS when the out-of-paper condition is detected. The SANE frontend scanadf(1) is a command line frontend that supports multi-page scans. It has been used successfully with this backend. The SANE frontend xsane(1) is an improved GUI frontend by Oliver Rauch. Support for multi-page scans is included in xsane version 0.35 and above.

### Duplex scanning

Some models, such as the COPISCAN II 6338, support duplex scanning. That is, they scan both sides of the document during a single pass through the scanner (the scanner has two cameras). This backend supports duplex scanning (with the --duplex option). The front and back page images are delivered consecutively as if they were



separately scanned pages.

#### Hardware compression

The scanner is capable of compressing the data into several industry standard formats (CCITT G3, CCITT G3-2D, CCITT G4). This results in increased performance as less data is passed from the scanner to the host over the SCSI bus. The backend supports these compression formats via the --g31d, --g32d, --g42d options, respectively. Many SANE frontends are not equipped to deal with these formats, however. The SANE frontend scanadf(1) supports these optional frame formats. The compressed image data is written directly to a file and can then be processed by a scan-script using the --scan-script option. Examples of this are given on the scanadf(1) homepage.

#### Automatic Border Detection

The scanner can automatically detect the paper size and adjust the scanning window geometry appropriately. The backend supports this useful feature with the --auto?border option. It is enabled by default.

#### Batch Mode Scanning

The batch scan mode allows for maximum throughput. The Set Window parameters must remain constant during the entire batch.

#### Icon Generation

The Icon function generates a thumbnail of the full page image, that can be transferred as if it were a separate page. This allows the host to quickly display a thumbnail representation during the scanning operation. Perhaps this would be a great way of implementing a preview scan, but since a normal scan is so quick, it might not be worth the trouble.

#### Multiple Sections

Multiple sections (scanning sub-windows) can be defined for the front and back pages. Each section can have different characteristics (e.g. geometry, compression). The sections are returned as if they were separately scanned images. Additionally sections can be used to greatly enhance the accuracy and efficiency of the barcode/patchcode decoding process by limiting the search area to a small subset of the page. Most Copiscan II series scanners support up to 8 user-defined sections.

#### Support Barcode/Patchcode Decoding

The RSC unit can recognize Bar and Patch Codes of various types embedded in the

scanned image. The codes are decoded and the data is returned to the frontend as a text frame. The text is encoded in xml and contains a great deal of information about the decoded data such as the location where it was found, its orientation, and the time it took to find. Further information on the content of this text frame as well as some barcode decoding examples can be found on the backend home? page.

## LIMITATIONS

### Decoding a single barcode type per scan

The RSC unit can search for up to six different barcode types at a time. While the code generally supports this as well, the `--barcode-search-bar` option only allows the user to specify a single barcode type. Perhaps another option which allows a comma separated list of barcode type codes could be added to address this.

### Scanning a fixed number of pages in batch mode

The separation of front and back end functionality in SANE presents a problem in supporting the 'cancel batch' functionality in the scanner. In batch mode, the scanner is always a page ahead of the host. The host, knowing ahead of time which page will be the last, can cancel batch mode prior to initiating the last scan command. Currently, there is no mechanism available for the frontend to pass this knowledge to the backend. If batch mode is enabled and the `--end-count` terminates a scanadf session, an extra page will be pulled through the scanner, but is neither read nor delivered to the frontend. The issue can be avoided by specifying `--batch=no` when scanning a fixed number of pages.

### Revision 1.2 Patch detector

There is an enhanced patchcode detection algorithm available in the RSC with revision 1.2 or higher that is faster and more reliable than the standard Bar/Patch code decoder. This is not currently supported.

## BUGS

This is a new backend; detailed bug reports are welcome -- and expected ;)

If you have found something that you think is a bug, please attempt to recreate it with the `SANE_DEBUG_BH` environment variable set to 255, and send a report detailing the conditions surrounding the bug to [sane-devel@alioth-lists.debian.net](mailto:sane-devel@alioth-lists.debian.net).

## SEE ALSO

[sane\(7\)](#), [sane-scsi\(5\)](#), [scanimage\(1\)](#), [scanadf\(1\)](#), [xsane\(1\)](#)

## AUTHOR

The sane-bh backend was written by Tom Martone, based on the sane-ricoh(5) backend by Federico W. Dillema and the bnhscan program by Sean Reifschneider of tummy.com ltd. Some 8000 enhancements added by Mark Temple.

10 Jul 2008

sane-bh(5)