NAME

sane−dll − SANE dynamic backend loader

DESCRIPTION

The **sane−dll** library implements a SANE (Scanner Access Now Easy) backend that provides access to an arbitrary number of other SANE backends. These backends may either be pre-loaded at the time the **sane−dll** library is built or, on systems that support dynamic loading of shared libraries, the backends may be loaded at runtime. In the latter case, adding support for a new backend simply involves installing the relevant library in *@LIBDIR@* and adding an entry to the *dll.conf* configuration file. In other words, no applications need to be modified or recompiled to add support for new devices.

DEVICE NAMES

This backend expects device names of the form:

> *backend*:*device*

Where *backend* is the name of the backend and *device* is the name of the device in this backend that should be addressed. If the device name does not contain a colon (:), then the entire string is treated as the *device* string for the default backend. The default backend is the backend listed last in the configuration file (see below) or the first pre-loaded backend (if any).

CONFIGURATION

The contents of the *dll.conf* file is a list of backend names that may be loaded dynamically upon demand. Empty lines are ignored, also everything after a hash mark (#). A sample configuration file is shown below:

```
net
# this is a comment
pnm
mustek
```

It is also possible to add a file in *@CONFIGDIR@/dll.d* that contains the list of backends to be added. Backends mentioned in a file included in this directory will be added before any backends listed in *dll.conf.* Files in *@CONFIGDIR@/dll.d* can be freely named. They shall follow the format conventions as apply for *dll.conf.*

Note that backends that were pre-loaded when building this library do not have to be listed in this configuration file. That is, if a backend was preloaded, then that backend will always be present, regardless of whether it's listed in the configuration file or not.

The list of preloaded backends is determined by macro **PRELOADABLE_BACKENDS** in file backend/Makefile.in of the SANE source code distribution. After changing the value of this macro, it is necessary to reconfigure, rebuild, and reinstall SANE for the change to take effect.

Aliases are defined in the config file *dll.aliases*. It can contain entries of the form

```
alias SomeName SaneDeviceName
alias "Some Name" SaneDeviceName
hide SaneDeviceName
```

For example:

```
alias Epson net:somehost:epson:/dev/sgX
alias "Siemens ST400" st400:/dev/sgY
hide net:somehost:pnm:0
hide net:somehost:pnm:1
alias "Read from file" pnm:0
hide pnm:1
```

Aliased device names are automatically hidden.

The idea is that users don’t have to deal with complicated device names (especially for networked devices), and to hide other exported devices which might confuse them. Note that a hidden device can still be accessed if the device name is known, it just doesn’t appear on the list.

FILES

@CONFIGDIR@/dll.aliases
The list of aliased or hidden backends.

@CONFIGDIR@/dll.conf
The backend configuration file (see also description of **SANE_CONFIG_DIR** below).

@LIBDIR@/libsane−dll.a
The static library implementing this backend.

@LIBDIR@/libsane−dll.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. Under UNIX, 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 @CONFIGDIR@. 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 "@CONFIGDIR@" being searched (in this order).

SANE_DEBUG_DLL
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.

```
Value  Description
0      print severe errors only
1      print normal errors and important messages
2      print normal messages
3      print debugging messages
4      print everything
```

Example: export SANE_DEBUG_DLL=3

SEE ALSO

sane(7), **scanimage**(1), **sane−"backendname"**(5)

AUTHOR

David Mosberger