

**NAME**

alsactl\_init - alsa control management - initialization

**DESCRIPTION**

"alsactl init" provides soundcard specific initialization.

**CONFIGURATION**

All "alsactl init" configuration files are placed in /usr/share/alsa/init/ directory. The top level configuration file is /usr/share/alsa/init/00main. The default top-level file can be also specified using -i or --initfile parameter for the alsactl tool. Every file consists of a set of lines of text. All empty lines or lines beginning with '#' will be ignored.

**Rules files**

The "alsactl init" rules are read from the files located in the /usr/share/alsa/init/\*. The top level configuration file is /usr/share/alsa/init/00main. Every line in the rules file contains at least one key value pair. There are two kind of keys, match and assignment keys. If all match keys are matching against its value, the rule gets applied and the assign keys get the specified value assigned.

A rule may consists of a list of one or more key value pairs separated by a comma. Each key has a distinct operation, depending on the used operator. Valid operators are:

==

Compare for equality.

!=

Compare for non-equality.

=

Assign a value to a key. Keys that represent a list, are reset and only this single value is assigned.

+=

Add the value to a key that holds a list of entries.

:=

Assign a value to a key finally; disallow any later changes, which may be used to prevent changes by any later rules.

The following key names can be used to match against device properties:

**CARDINDEX**

Match the card index of the ALSA driver.

**CTL{attribute}**

Set or test universal control attribute. Possible attributes:

**numid**

Numeric control identification.

**iface, interface**

Control interface name (CARD, HWEDEP, MIXER, PCM, RAWMIDI, TIMER, SEQUENCER)

**subdev, subdevice**

Subdevice number.

**name**

Control name

**index**

Control index

**type**

Control type (BOOLEAN, INTEGER, INTEGER64, ENUMERATED, BYTES, IEC958)

**attr, attribute**

Attributes (stored in a string - use match characters \* and ?):

**r** control is readable

**w** control is writable

**v** control is volatile

**i** control is inactive

**l** control is locked

**R** control is TLV readable

**W** control is TLV writable

**C** control is TLV commandable

**o** process is owner of this control

**u** control created in user space

**owner**

Control owner process PID number

**count**

Control count of values

**min**

Value range - minimum value

**max**

Value range - maximum value

**step**

Value range - step value

**dBmin**

Value range - minimum dB value

**dBmax**

Value range - maximum dB value

**items**

Enumerated value - number of text items

**enums**

Enumerated value - list of text names stored between '|' character

**value**

Value of control stored to a string delimited by comma (,).

**do\_search**

Search for a control. Value "1" is returned if a control was found. The CTL{name} key might contain match characters \* and ?. An control index might be specified as first argument starting from zero (e.g. CTL{do\_search 2}="1").

**do\_count**

Search for a controls and return total count of matched ones. The CTL{name} key might contain

match characters \* and ?.

### **CONFIG{sysfs\_device}**

The relative path to sysfs subsystem specifying the root directory of a soundcard device. Usually, it should be set to "/class/sound/card\$cardinfo{card}/device".

### **ATTR{filename}**

Match sysfs attribute values of the soundcard device. The relative path to sysfs tree must be defined by CONFIG{sysfs\_device} key. Trailing whitespace in the attribute values is ignored, if the specified match value does not contain trailing whitespace itself. Depending on the type of operator, this key is also used to set the value of a sysfs attribute.

### **ENV{key}**

Match against the value of an environment variable. Up to five **ENV** keys can be specified per rule. Depending on the type of operator, this key is also used to export a variable to the environment.

### **PROGRAM**

Execute external program. The key is true, if the program returns without exit code zero. The whole event environment is available to the executed program. The program's output printed to stdout is available for the RESULT key.

Several builtin commands are available:

#### **\_\_ctl\_search**

Search for a control. The CTL{name} key might contain match characters \* and ?. An control index might be specified as first argument starting from zero (e.g. PROGRAM="\_\_ctl\_search 2").

#### **\_\_ctl\_count**

Search for a controls and return total count of matched ones. The CTL{name} key might contain match characters \* and ?.

### **RESULT**

Match the returned string of the last PROGRAM call. This key can be used in the same or in any later rule after a PROGRAM call.

Most of the fields support a shell style pattern matching. The following pattern characters are supported:

\*

Matches zero, or any number of characters.

?

Matches any single character.

[]

Matches any single character specified within the brackets. For example, the pattern string 'tty[SR]' would match either 'ttyS' or 'ttyR'. Ranges are also supported within this match with the '-' character. For example, to match on the range of all digits, the pattern [0-9] would be used. If the first character following the '[' is a '!', any characters not enclosed are matched.

The following keys can get values assigned:

### **CTL{numid}, CTL{iface}, CTL{device}, CTL{subdev}, CTL{name}, CTL{index},**

Select universal control element.

### **CTL{value}**

Value is set (written) also to soundcard's control device and RESULT key is set to errno code. The result of set operation is always true (it means continue with next key on line).

### **CTL{values}**

Value is set (written) also to soundcard's control device (all control values are set to specified value) and RESULT key is set to errno code. The result of set operation is always true (it means continue with next key on line).

### **CTL{write}**

Value is set (written) also to soundcard's control device (all control values are set to specified value).

The result of set operation is true when operation succeed (it means continue with next key on line).

**ENV{key}**

Export a variable to the environment. Depending on the type of operator, this key is also to match against an environment variable.

**RESULT**

Set RESULT variable. Note that PROGRAM also sets this variable, but setting this variable manually might be useful to change code execution order (included files).

**LABEL**

Named label where a GOTO can jump to.

**GOTO**

Jumps to the next LABEL with a matching name. The goto cannot jump backward.

**INCLUDE**

Include the specified filename or files in specified directory.

When a directory is specified, only the files with the extension ".conf" are read. Also they are read in the alphabetical order. Thus it's highly recommended to use some number prefix (e.g. "01-something.conf") to assure the order of executions.

**ACCESS**

Check if specified file or directory exists

**CONFIG{sysfs\_device}**

The relative path to sysfs subsystem specifying the root directory of a soundcard device. Usually, it should be set to "/class/sound/card\$cardinfo{card}/device".

**PRINT**

PRINT value to stdout.

**ERROR**

PRINT value to stderr.

**EXIT**

Exit immediately and set program exit code to value (should be integer). If value is "return" string, parser leaves current included file and returns to parent configuration file.

The **PROGRAM**, **RESULT**, **CTL{value}**, **PRINT**, **ERROR**, **EXIT**, **CONFIG{}** fields support simple printf-like string substitutions. It allows the use of the complete environment set by earlier matching rules. For all other fields, substitutions are applied while the individual rule is being processed. The available substitutions are:

**\$cardinfo{attribute}, %i{attribute}**

See CARDINFO{ } for more details.

**\$ctl{attribute}, %C{attribute}**

See CTL{ } for more details.

**\$attr{file}, %s{file}**

The value of a sysfs attribute found at the device, where all keys of the rule have matched. If the attribute is a symlink, the last element of the symlink target is returned as the value.

**\$env{key}, %E{key}**

The value of an environment variable.

**\$result, %c**

The string returned by the external program requested with PROGRAM. A single part of the string, separated by a space character may be selected by specifying the part number as an attribute: **%c{N}**. If the number is followed by the '+' char this part plus all remaining parts of the result string are substituted: **%c{N+}**

**\$sysfsroot, %r**

Root directory where sysfs file-system is mounted. Usually, this value is just `"/sys"`.

**\$config{key}, %g{key}**

The value of a configuration variable. See `CONFIG{ }` for more details.

**%%**

The `'%'` character itself.

**\$\$**

The `'$'` character itself.

The count of characters to be substituted may be limited by specifying the format length value. For example, `%3s{file}` will only insert the first three characters of the `sysfs` attribute

## **AUTHOR**

Written by Jaroslav Kysela <perex@perex.cz>

Some portions are written by Greg Kroah-Hartman <greg@kroah.com> and Kay Sievers <kay.sievers@vrfy.org>.

## **SEE ALSO**

[alsactl\(1\)](#)