

NAME

sysrc - safely edit system rc files

SYNOPSIS

sysrc [-**cdDeEFhinNqv**x] [-**s** *name*] [-**f** *file*] [-**j** *jail* | -**R** *dir*] *name*[[+|=]*value*] ...

sysrc [-**cdDeEFhinNqv**x] [-**s** *name*] [-**f** *file*] [-**j** *jail* | -**R** *dir*] -**a** | -**A**

sysrc [-**E**] [-**s** *name*] [-**f** *file*] -**I**

sysrc [-**eEqv**] -**L** [*name* ...]

DESCRIPTION

The **sysrc** utility retrieves rc.conf(5) variables from the collection of system rc files and allows processes with appropriate privilege to change values in a safe and effective manner.

The following options are available:

- a** Dump a list of all non-default configuration variables.
- A** Dump a list of all configuration variables (incl. defaults).
- c** Check only. For querying, return success if all requested variables are set (even if NULL), otherwise return error status. For assignments, return success if no changes are required, otherwise failure. If verbose (see "-**v**") prints a message stating whether variables are set and/or changes are required.
- d** Print a description of the given variable.
- D** Show default value(s) only (this is the same as setting RC_CONFS to NULL or passing '-**f**' with a NULL file-argument).
- e** Print query results as sh(1) compatible syntax (for example, 'var=value'). Ignored if either '-**n**' or '-**F**' is specified.
- E** When given '-**I**' or '-**L**' to list configuration files, only list those that exist. When changing a setting, prefer to modify existing files.
- f** *file* Operate on the specified file(s) instead of the files obtained by reading the 'rc_conf_files' entry in the RC_DEFAULTS file. This option can be specified multiple times for additional files.
- F** Show only the last rc.conf(5) file each directive is in.

- h** Print a short usage message to stderr and exit.
- help** Print a full usage statement to stderr and exit.
- i** Ignore unknown variables.
- j jail** The *jid* or name of the *jail* to operate within (overrides '**-R dir**'); requires `jexec(8)`).
- l** List configuration files used at startup on stdout and exit.
- L** List all configuration files including `rc.conf.d` entries on stdout and exit. Can be combined with '**-v**' or '**-e**' to show service names. `sysrc` exits with success if all named services are installed, failure otherwise.
- n** Show only variable values, not their names.
- N** Show only variable names, not their values.
- q** Quiet. Disable verbose and hide certain errors. When combined with '**-L**' and one or more *name* arguments, provide only exit status and no output.
- R dir** Operate within the root directory '*dir*' rather than '/'.

-s name
If an `rc.d` script of *name* exists (in `/etc/rc.d` or `local_startup` directories), process its `rc.conf.d` entries as potential overrides to '`rc_conf_files`'. See `rc.subr(8)` for additional information on `rc.conf.d`. Can be combined with '**-l**' to list configuration files used by service at startup.
- v** Verbose. Print the pathname of the specific `rc.conf(5)` file where the directive was found.
- version**
Print version information to stdout and exit.
- x** Remove variable(s) from specified file(s).

This utility has a similar syntax to `sysctl(8)`. It shares the '**-e**' and '**-n**' options (detailed above) and also has the same '`name[=value]`' syntax for making queries/assignments. In addition (but unlike `sysctl(8)`), '`name+=value`' is supported for adding items to values (see APPENDING VALUES) and '`name-=value`' is supported for removing items from values (see SUBTRACTING VALUES).

However, while `sysctl(8)` serves to query/modify MIBs in the entrant kernel, **sysrc** instead works on values in the system `rc.conf(5)` configuration files.

The list of system configuration files is configured in the file `/etc/defaults/rc.conf` within the variable `'rc_conf_files'`, which by-default contains a space-separated list of pathnames. On all FreeBSD systems, this defaults to the value `"/etc/rc.conf /etc/rc.conf.local"`. Each pathname is sourced in-order upon startup. It is in the same fashion that **sysrc** sources the configuration files before returning the value of the given variable.

When supplied a variable name, **sysrc** will return the value of the variable. If the variable does not appear in any of the configured `'rc_conf_files'`, an error is printed and error status is returned.

When changing values of a given variable, it does not matter if the variable appears in any of the `'rc_conf_files'` or not. If the variable does not appear in any of the files, it is appended to the end of the first pathname in the `'rc_conf_files'` variable. Otherwise, **sysrc** will replace only the last-occurrence in the last-file found to contain the variable. This gets the value to take effect next boot without heavily modifying these integral files (yet taking care not to allow the file to grow unwieldy should **sysrc** be called repeatedly).

APPENDING VALUES

When using the `'key+=value'` syntax to add items to existing values, the first character of the value is taken as the delimiter separating items (usually `" "` or `","`). For example, in the following statement:

```
sysrc cloned_interfaces+=" gif0"
```

the first character is a space, informing **sysrc** that existing values are to be considered separated by whitespace. If `'gif0'` is not found in the existing value for `cloned_interfaces`, it is added (with delimiter only if existing value is non-NULL).

For convenience, if the first character is alpha-numeric (letters A-Z, a-z, or numbers 0-9), dot (`.`), or slash (`/`), **sysrc** uses the default setting of whitespace as separator. For example, the above and below statements are equivalent since `"gif0"` starts with an alpha-numeric character (the letter `g`):

```
sysrc cloned_interfaces+=gif0
```

Take the following sequence for example:

```
sysrc cloned_interfaces= # start with NULL
```

```
sysrc cloned_interfaces+=gif0
```

```
# NULL -> 'gif0' (NB: no preceding delimiter)
```

```
sysrc cloned_interfaces+=gif0 # no change
```

```
sysrc cloned_interfaces+="tun0 gif0"
```

```
# 'gif0' -> 'gif0 tun0' (NB: no duplication)
```

sysrc prevents the same value from being added if already there.

SUBTRACTING VALUES

When using the 'key-=value' syntax to remove items from existing values, the first character of the value is taken as the delimiter separating items (usually " " or ","). For example, in the following statement:

```
cloned_interfaces-=" gif0"
```

the first character is a space, informing **sysrc** that existing values are to be considered separated by whitespace. If 'gif0' is found in the existing value for *cloned_interfaces*, it is removed (extra delimiters removed).

For convenience, if the first character is alpha-numeric (letters A-Z, a-z, or numbers 0-9), dot (.), or slash (/), **sysrc** uses the default setting of whitespace as separator. For example, the above and below statements are equivalent since "gif0" starts with an alpha-numeric character (the letter g):

```
sysrc cloned_interfaces-=gif0
```

Take the following sequence for example:

```
sysrc foo="bar baz" # start
```

```
sysrc foo-=bar # 'bar baz' -> 'baz'
```

```
sysrc foo-=baz # 'baz' -> NULL
```

sysrc removes all occurrences of all items provided and collapses extra delimiters between items.

ENVIRONMENT

The following environment variables are referenced by **sysrc**:

RC_CONFS Override default 'rc_conf_files' (even if set to NULL).

RC_DEFAULTS Location of `/etc/defaults/rc.conf` file.

DEPENDENCIES

The following standard commands are required by **sysrc**:

awk(1), cat(1), chmod(1), env(1), grep(1), mktemp(1), mv(1), rm(1), sh(1), stat(1), tail(1), chown(8), chroot(8), jls(8), and jexec(8).

FILES

/etc/defaults/rc.conf
/etc/rc.conf
/etc/rc.conf.local
/etc/rc.conf.d/name
*/etc/rc.conf.d/name/**
/usr/local/etc/rc.conf.d/name
*/usr/local/etc/rc.conf.d/name/**

EXAMPLES

Below are some simple examples of how **sysrc** can be used to query certain values from the `rc.conf(5)` collection of system configuration files:

sysrc sshd_enable
returns the value of `$sshd_enable`, usually YES or NO.

sysrc defaultrouter
returns IP address of default router (if configured).

Working on other files, such as `crontab(5)`:

sysrc -f /etc/crontab MAILTO
returns the value of the MAILTO setting (if configured).

Appending to existing values:

sysrc cloned_interfaces+=gif0
appends "gif0" to `$cloned_interfaces` (see APPENDING VALUES).

sysrc cloned_interfaces-=gif0
removes "gif0" from `$cloned_interfaces` (see SUBTRACTING VALUES).

In addition to the above syntax, **sysrc** also supports inline sh(1) PARAMETER expansion for changing the way values are reported, shown below:

sysrc 'hostname%%.*'

returns \$hostname up to (but not including) first '.'.

sysrc 'network_interfaces%%[\$IFS]*'

returns first word of \$network_interfaces.

sysrc 'ntpdate_flags##*[\$IFS]'

returns last word of \$ntpdate_flags (time server address).

sysrc usbd_flags-"default"

returns \$usbd_flags or default if unset or NULL.

sysrc cloned_interfaces+"alternate"

returns alternate if \$cloned_interfaces is set.

SEE ALSO

rc.conf(5), jail(8), jexec(8), jls(8), rc(8), rc.subr(8), sysctl(8)

HISTORY

A **sysrc** utility first appeared in FreeBSD 9.2.

AUTHORS

Devin Teske <dteske@FreeBSD.org>

THANKS TO

Brandon Gooch, Enji Cooper, Julian Elischer, Pawel Jakub Dawidek, Cyrille Lefevre, Ross West, Stefan Esser, Marco Steinbach, Jilles Tjoelker, Allan Jude, and Lars Engels for suggestions, help, and testing.