NAME

sysctl - get or set kernel state

SYNOPSIS

sysctl [**-bdeFhilNnoTtqWx**] [**-B** *bufsize*] [**-f** *filename*] *name*[=*value*[,*value*]] ...
sysctl [**-bdeFhlNnoTtqWx**] [**-B** *bufsize*] **-a**

DESCRIPTION

The **sysctl** utility retrieves kernel state and allows processes with appropriate privilege to set kernel state. The state to be retrieved or set is described using a "Management Information Base" ("MIB") style name, described as a dotted set of components.

The following options are available:

-A Equivalent to **-o -a** (for compatibility).

-a List all the currently available values except for those which are opaque or excluded from listing via the CTLFLAG_SKIP flag. This option is ignored if one or more variable names are specified on the command line.

-b Force the value of the variable(s) to be output in raw, binary format. No names are printed and no terminating newlines are output. This is mostly useful with a single variable.

-B *bufsize*
Set the buffer size to read from the **sysctl** to *bufsize*. This is necessary for a **sysctl** that has variable length, and the probe value of 0 is a valid length, such as *kern.arandom*.

-d Print the description of the variable instead of its value.

-e Separate the name and the value of the variable(s) with '='. This is useful for producing output which can be fed back to the **sysctl** utility. This option is ignored if either **-N** or **-n** is specified, or a variable is being set.

-f *filename*
Specify a file which contains a pair of name and value in each line. **sysctl** reads and processes the specified file first and then processes the name and value pairs in the command line argument.

-F Print the format of the variable. This is additional information to describe the type of the variable and most useful with struct types such as clockinfo, timeval, and loadavg.

-h Format output for human, rather than machine, readability.

-i Ignore unknown OIDs. The purpose is to make use of **sysctl** for collecting data from a variety of machines (not all of which are necessarily running exactly the same software) easier.

-l Show the length of variables along with their values. This option cannot be combined with the **-N** option.

-N Show only variable names, not their values. This is particularly useful with shells that offer programmable completion. To enable completion of variable names in zsh(1) (*ports/shells/zsh*), use the following code:

```
listsysctls () { set -A reply $(sysctl -AN ${1%.*}) }
compctl -K listsysctls sysctl
```

To enable completion of variable names in tcsh(1), use:

```
complete sysctl 'n/*/`sysctl -Na`/'
```

-n Do not show variable names. This option is useful for setting shell variables. For instance, to save the pagesize in variable *psize*, use:

```
set psize=`sysctl -n hw.pagesize`
```

-o Show opaque variables (which are normally suppressed). The format and length are printed, as well as a hex dump of the first sixteen bytes of the value.

-q Suppress some warnings generated by **sysctl** to standard error.

-T Display only variables that are settable via loader (CTLFLAG_TUN).

-t Print the type of the variable.

-W Display only writable variables that are not statistical. Useful for determining the set of runtime tunable sysctls.

-X Equivalent to **-x -a** (for compatibility).

-x As **-o**, but prints a hex dump of the entire value instead of just the first few bytes.

The information available from **sysctl** consists of integers, strings, and opaque types. The **sysctl** utility only knows about a couple of opaque types, and will resort to hexdumps for the rest. The opaque information is much more useful if retrieved by special purpose programs such as ps(1), systat(1), and netstat(1).

Some of the variables which cannot be modified during normal system operation can be initialized via loader(8) tunables. This can for example be done by setting them in loader.conf(5). Please refer to loader.conf(5) for more information on which tunables are available and how to set them.

The string and integer information is summarized below. For a detailed description of these variables see sysctl(3) and security(7).

The changeable column indicates whether a process with appropriate privilege can change the value. String and integer values can be set using **sysctl**.

Name	Type	Changeable
kern.ostype	string	no
kern.osrelease	string	no
kern.osrevision	integer	no
kern.version	string	no
kern.maxvnodes	integer	yes
kern.maxproc	integer	no
kern.maxprocperuid	integer	yes
kern.maxfiles	integer	yes
kern.maxfilesperproc	integer	yes
kern.argmax	integer	no
kern.securelevel	integer	raise only
kern.hostname	string	yes
kern.hostid	integer	yes
kern.clockrate	struct	no
kern.posix1version	integer	no
kern.ngroups	integer	no
kern.job_control	integer	no
kern.saved_ids	integer	no
kern.boottime	struct	no
kern.domainname	string	yes
kern.filedelay	integer	yes
kern.dirdelay	integer	yes
kern.metadelay	integer	yes
kern.osreldate	integer	no

kern.bootfile	string	yes
kern.corefile	string	yes
kern.logsigexit	integer	yes
security.bsd.suser_enabled	integer	yes
security.bsd.see_other_uids	integer	yes
security.bsd.see_other_gids	integer	yes
security.bsd.see_jail_proc	integer	yes
security.bsd.unprivileged_proc_debug	integer	yes
security.bsd.unprivileged_read_msgbuf	integer	yes
vm.loadavg	struct	no
hw.machine	string	no
hw.model	string	no
hw.ncpu	integer	no
hw.byteorder	integer	no
hw.physmem	integer	no
hw.usermem	integer	no
hw.pagesize	integer	no
hw.floatingpoint	integer	no
hw.machine_arch	string	no
hw.realmem	integer	no
machdep.adjkerntz	integer	yes
machdep.disable_rtc_set	integer	yes
machdep.guessed_bootdev	string	no
user.cs_path	string	no
user.bc_base_max	integer	no
user.bc_dim_max	integer	no
user.bc_scale_max	integer	no
user.bc_string_max	integer	no
user.coll_weights_max	integer	no
user.expr_nest_max	integer	no
user.line_max	integer	no
user.re_dup_max	integer	no
user.posix2_version	integer	no
user.posix2_c_bind	integer	no
user.posix2_c_dev	integer	no
user.posix2_char_term	integer	no
user.posix2_fort_dev	integer	no
user.posix2_fort_run	integer	no
user.posix2_localedef	integer	no
user.posix2_sw_dev	integer	no

user.posix2_upe	integer	no
user.stream_max	integer	no
user.tzname_max	integer	no
user.localbase	string	no

FILES

<*sys/sysctl.h*> definitions for top level identifiers, second level kernel and hardware identifiers, and user level identifiers

<*sys/socket.h*> definitions for second level network identifiers

<*sys/gmon.h*> definitions for third level profiling identifiers

<*vm/vm_param.h*>
definitions for second level virtual memory identifiers

<*netinet/in.h*> definitions for third level Internet identifiers and fourth level IP identifiers

<*netinet/icmp_var.h*>
definitions for fourth level ICMP identifiers

<*netinet/udp_var.h*>
definitions for fourth level UDP identifiers

EXIT STATUS

The **sysctl** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

For example, to retrieve the maximum number of processes allowed in the system, one would use the following request:

```
sysctl kern.maxproc
```

To set the maximum number of processes allowed per uid to 1000, one would use the following request:

```
sysctl kern.maxprocperuid=1000
```

Information about the system clock rate may be obtained with:

```
sysctl kern.clockrate
```

Information about the load average history may be obtained with:

```
sysctl vm.loadavg
```

More variables than these exist, and the best and likely only place to search for their deeper meaning is

undoubtedly the source where they are defined.

COMPATIBILITY

The **-w** option has been deprecated and is silently ignored.

SEE ALSO

sysctl(3), loader.conf(5), sysctl.conf(5), security(7,) loader(8)

HISTORY

A **sysctl** utility first appeared in 4.4BSD.

In FreeBSD 2.2, **sysctl** was significantly remodeled.

BUGS

The **sysctl** utility presently exploits an undocumented interface to the kernel sysctl(9) facility to traverse the sysctl tree and to retrieve format and name information. This correct interface is being thought about for the time being.