

**NAME**

**rctl\_add\_rule**, **rctl\_get\_limits**, **rctl\_get\_racct**, **rctl\_get\_rules**, **rctl\_remove\_rule** - manipulate and query the resource limits database

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <sys/rctl.h>
```

*int*

```
rctl_add_rule(const char *inbufp, size_t inbuflen, char *outbufp, size_t outbuflen);
```

*int*

```
rctl_get_limits(const char *inbufp, size_t inbuflen, char *outbufp, size_t outbuflen);
```

*int*

```
rctl_get_racct(const char *inbufp, size_t inbuflen, char *outbufp, size_t outbuflen);
```

*int*

```
rctl_get_rules(const char *inbufp, size_t inbuflen, char *outbufp, size_t outbuflen);
```

*int*

```
rctl_remove_rule(const char *inbufp, size_t inbuflen, char *outbufp, size_t outbuflen);
```

**DESCRIPTION**

These system calls are used to manipulate and query the resource limits database. For all functions, *inbuflen* refers to the length of the buffer pointed to by *inbufp* and *outbuflen* refers to the length of the buffer pointed to by *outbufp*.

The **rctl\_add\_rule()** function adds the rule pointed to by *inbufp* to the resource limits database. The *outbufp* and *outbuflen* arguments are unused. Rule format is as described in rctl(8), with exceptions noted in the *RULES AND FILTERS* section.

The **rctl\_get\_limits()** function returns in *outbufp* a comma-separated list of rules that apply to the process that matches the filter specified in *inbufp*. This includes rules with a subject of the process itself as well as rules with a different subject (such as user or loginclass) that apply to the process.

The **rctl\_get\_racct()** function returns resource usage information for a given subject. The subject is specified by passing a filter in *inbufp*. Filter syntax is as described in rctl(8), with exceptions noted in

the *RULES AND FILTERS* section. A comma-separated list of resources and the amount used of each by the specified subject is returned in *outbufp*. The resource and amount is formatted as "resource=amount".

The **rctl\_get\_rules()** function returns in *outbufp* a comma-separated list of rules from the resource limits database that match the filter passed in *inbufp*. Filter syntax is as described in `rctl(8)`, with exceptions noted in the *RULES AND FILTERS* section. A filter of `::` may be passed to return all rules.

The **rctl\_remove\_rule()** function removes all rules matching the filter passed in *inbufp* from the resource limits database. Filter syntax is as described in `rctl(8)`, with exceptions noted in the *RULES AND FILTERS* section. *outbufp* and *outbuflen* are unused.

## RULES AND FILTERS

This section explains how the rule and filter format described in `rctl(8)` differs from the format passed to the system calls themselves. The `rctl` tool provides several conveniences that the system calls do not.

When using the system call:

- The subject must be fully specified. For example, abbreviating 'user' to 'u' is not acceptable.
- User and group IDs must be numeric. For example, 'root' must be expressed as '0'.
- Units are not permitted on resource amounts. For example, a quantity of 1024 bytes must be expressed as '1024' and not '1k'.

## RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

## ERRORS

The `rctl` system calls may fail if:

[ENOSYS]	RACCT/RCTL support is not present in the kernel or the <i>kern.racct.enable</i> sysctl is 0.
[EINVAL]	The rule or filter passed in <i>inbufp</i> is invalid.
[EPERM]	User has insufficient privileges to carry out the requested operation.
[E2BIG]	<i>inbufp</i> or <i>outbufp</i> are too large.

- [ESRCH] No process matched the provided rule or filter.
- [ENAMETOOLONG] The loginclass or jail name specified is too long.
- [ERANGE] The rule amount is outside of the allowable range or *outbufp* is too small.
- [EOPNOTSUPP] The requested operation is not supported for the given rule or filter.
- [EFAULT] *inbufp* or *outbufp* refer to invalid addresses.

**SEE ALSO**

rctl(8)

**HISTORY**

The rctl family of system calls appeared in FreeBSD 9.0.

**AUTHORS**

The rctl system calls were developed by Edward Tomasz Napierala <[trasz@FreeBSD.org](mailto:trasz@FreeBSD.org)> under sponsorship from the FreeBSD Foundation. This manual page was written by Eric Badger <[badger@FreeBSD.org](mailto:badger@FreeBSD.org)>.