```
**********************************************************
   12271 Wed Aug  3 14:06:32 2016
new/usr/src/man/man3c/rctlblk_set_value.3c
7264 Example code is rctlblk_set_value(3c) manpage does not compile.
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Toomas Soome <tsoome@me.com>
**********************************************************
'\" te
.\" Copyright (c) 2006, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH RCTLBLK_SET_VALUE 3C "Aug 2, 2016"
.TH RCTLBLK_SET_VALUE 3C "May 15, 2006"
.SH NAME
rctlblk_set_value, rctlblk_get_firing_time, rctlblk_get_global_action,
rctlblk_get_global_flags, rctlblk_get_local_action, rctlblk_get_local_flags,
rctlblk_get_privilege, rctlblk_get_recipient_pid, rctlblk_get_value,
rctlblk_get_enforced_value, rctlblk_set_local_action, rctlblk_set_local_flags,
rctlblk_set_privilege, rctlblk_set_recipient_pid, rctlblk_size \- manipulate
resource control blocks
.SH SYNOPSIS
.LP
.nf
#include <rctl.h>

\fBhrtime_t\fR  \fBrctlblk_get_firing_time\fR(\fBrctlblk_t *\fR\fIrblk\fR);
.fi

.LP
.nf
\fBint\fR \fBrctlblk_get_global_action\fR(\fBrctlblk_t *\fR\fIrblk\fR);
.fi

.LP
.nf
\fBint\fR \fBrctlblk_get_global_flags\fR(\fBrctlblk_t *\fR\fIrblk\fR);
.fi

.LP
.nf
\fBint\fR \fBrctlblk_get_local_action\fR(\fBrctlblk_t *\fR\fIrblk\fR, \fBint *\f
.fi

.LP
.nf
\fBint\fR \fBrctlblk_get_local_flags\fR(\fBrctlblk_t *\fR\fIrblk\fR);
.fi

.LP
.nf
\fBrctl_priv_t\fR  \fBrctlblk_get_privilege\fR(\fBrctlblk_t *\fR\fIrblk\fR);
.fi

.LP
.nf
\fBid_t\fR \fBrctlblk_get_recipient_pid\fR(\fBrctlblk_t *\fR\fIrblk\fR);
.fi

.LP
.nf
\fBrctl_qty_t\fR  \fBrctlblk_get_value\fR(\fBrctlblk_t *\fR\fIrblk\fR);
.fi

.LP
.nf
```

```
\fBrctl_qty_t\fR  \fBrctlblk_get_enforced_value\fR(\fBrctlblk_t *\fR\fIrblk\fR);
.fi

.LP
.nf
\fBvoid\fR \fBrctlblk_set_local_action\fR(\fBrctlblk_t *\fR\fIrblk\fR, \fBrctl_a
     \fBint\fR \fIsignal\fR);
.fi

.LP
.nf
\fBvoid\fR \fBrctlblk_set_local_flags\fR(\fBrctlblk_t *\fR\fIrblk\fR, \fBint\fR
.fi

.LP
.nf
\fBvoid\fR \fBrctlblk_set_privilege\fR(\fBrctlblk_t *\fR\fIrblk\fR, \fBrctl_priv
.fi

.LP
.nf
\fBvoid\fR \fBrctlblk_set_value\fR(\fBrctlblk_t *\fR\fIrblk\fR, \fBrctl_qty_t\fR
.fi

.LP
.nf
\fBvoid\fR  \fBrctlblk_set_recipient_pid\fR(\fBid_t\fR\fIpid\fR);
.fi

.LP
.nf
\fBsize_t\fR \fBrctlblk_size\fR(\fBvoid\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The resource control block routines allow the establishment or retrieval of
values from a resource control block used to transfer information using the
\fBgetrctl\fR(2) and \fBsetrctl\fR(2) functions. Each of the routines accesses
or sets the resource control block member corresponding to its name.  Certain
of these members are read-only and do not possess set routines.
.sp
.LP
The firing time of a resource control block is 0 if the resource control
action-value has not been exceeded for its lifetime on the process.  Otherwise
the firing time is the value of \fBgethrtime\fR(3C) at the moment the action on
the resource control value was taken.
.sp
.LP
The global actions and flags are the action and flags set by \fBrctladm\fR(1M).
These values cannot be set with \fBsetrctl\fR(2).  Valid global actions are
listed in the table below. Global flags are generally a published property of
the control and are not modifiable.
.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_DENY_ALWAYS\fR\fR
.ad
.RS 28n
The action taken when a control value is exceeded on this control will always
include denial of the resource.
.RE

.sp
.ne 2
```

```
.na
\fB\fBRCTL_GLOBAL_DENY_NEVER\fR\fR
.ad
.RS 28n
The action taken when a control value is exceeded on this control will always
exclude denial of the resource; the resource will always be granted, although
other actions can also be taken.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_SIGNAL_NEVER\fR\fR
.ad
.RS 28n
No signal actions are permitted on this control.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_CPU_TIME\fR\fR
.ad
.RS 28n
The valid signals available as local actions include the \fBSIGXCPU\fR signal.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_FILE_SIZE\fR\fR
.ad
.RS 28n
The valid signals available as local actions include the \fBSIGXFSZ\fR signal.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_INFINITE\fR\fR
.ad
.RS 28n
This resource control supports the concept of an unlimited value; generally
true only of accumulation-oriented resources, such as CPU time.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_LOWERABLE\fR\fR
.ad
.RS 28n
Non-privileged callers are able to lower the value of privileged resource
control values on this control.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_NOACTION\fR\fR
.ad
.RS 28n
No global action will be taken when a resource control value is exceeded on
this control.
.RE
```

```
.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_NOBASIC\fR\fR
.ad
.RS 28n
No values with the \fBRCPRIV_BASIC\fR privilege are permitted on this control.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_SYSLOG\fR\fR
.ad
.RS 28n
A standard message will be logged by the \fBsyslog\fR(3C) facility when any
resource control value on a sequence associated with this control is exceeded.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_SYSLOG_NEVER\fR\fR
.ad
.RS 28n
The resource control does not support the \fBsyslog()\fR global action.
Exceeding a resource control value on this control will not result in a message
logged by the \fBsyslog()\fR facility.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_UNOBSERVABLE\fR\fR
.ad
.RS 28n
The resource control (generally on a task- or project-related control) does not
support observational control values. An \fBRCPRIV_BASIC\fR privileged control
value placed by a process on the task or process will generate an action only
if the value is exceeded by that process.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_BYTES\fR\fR
.ad
.RS 28n
This resource control represents a number of bytes.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_SECONDS\fR\fR
.ad
.RS 28n
This resource control represents a quantity of time in seconds.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_GLOBAL_COUNT\fR\fR
.ad
.RS 28n
```

```
This resource control represents an integer count.
.RE

.sp
.LP
The local action and flags are those on the current resource control value
represented by this resource control block. Valid actions and flags are listed
in the table below. In the case of \fBRCTL_LOCAL_SIGNAL\fR, the second argument
to \fBrctlblk_set_local_action()\fR contains the signal to be sent. Similarly,
the signal to be sent is copied into the integer location specified by the
second argument to \fBrctlblk_get_local_action()\fR. A restricted set of
signals is made available for normal use by the resource control facility:
\fBSIGBART\fR, \fBSIGXRES\fR, \fBSIGHUP\fR, \fBSIGSTOP\fR, \fBSIGTERM\fR, and
\fBSIGKILL\fR. Other signals are permitted due to global properties of a
specific control. Calls to \fBsetrctl()\fR with illegal signals will fail.
.sp
.ne 2
.na
\fB\fBRCTL_LOCAL_DENY\fR\fR
.ad
.RS 23n
When this resource control value is encountered, the request for the resource
will be denied. Set on all values if \fBRCTL_GLOBAL_DENY_ALWAYS\fR is set for
this control; cleared on all values if \fBRCTL_GLOBAL_DENY_NEVER\fR is set for
this control.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_LOCAL_MAXIMAL\fR\fR
.ad
.RS 23n
This resource control value represents a request for the maximum amount of
resource for this control. If \fBRCTL_GLOBAL_INFINITE\fR is set for this
resource control, \fBRCTL_LOCAL_MAXIMAL\fR indicates an unlimited resource
control value, one that will never be exceeded.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_LOCAL_NOACTION\fR\fR
.ad
.RS 23n
No local action will be taken when this resource control value is exceeded.
.RE

.sp
.ne 2
.na
\fB\fBRCTL_LOCAL_SIGNAL\fR\fR
.ad
.RS 23n
The specified signal, sent by \fBrctlblk_set_local_action()\fR, will be sent to
the process that placed this resource control value in the value sequence. This
behavior is also true for signal actions on project and task resource controls.
The specified signal is sent only to the recipient process, not all processes
within the project or task.
.RE

.sp
.LP
The \fBrctlblk_get_recipient_pid()\fR function returns the value of the process
ID that placed the resource control value for basic rctls. For privileged or
system rctls, \fBrctlblk_get_recipient_pid()\fR returns -1.
```

```
.sp
.LP
The \fBrctlblk_set_recipient_pid()\fR function sets the recipient \fIpid\fR for
a basic rctl. When \fBsetrctl\fR(2) is called with the flag
\fBRCTL_USE_RECIPIENT_PID\fR, this \fIpid\fR is used. Otherwise, the PID of the
calling process is used. Only privileged users can set the recipient PID to one
other than the PID of the calling process.  Process-scoped rctls must have a
recipient PID that matches the PID of the calling process.
.sp
.LP
The \fBrctlblk_get_privilege()\fR function returns the privilege of the
resource control block. Valid privileges are \fBRCPRIV_BASIC\fR,
\fBRCPRIV_PRIVILEGED\fR, and \fBRCPRIV_SYSTEM\fR. System resource controls are
read-only. Privileged resource controls require the {\fBPRIV_SYS_RESOURCE\fR}
privilege to write, unless the \fBRCTL_GLOBAL_LOWERABLE\fR global flag is set,
in which case unprivileged applications can lower the value of a privileged
control.
.sp
.LP
The \fBrctlblk_get_value()\fR and \fBrctlblk_set_value()\fR functions return or
establish the enforced value associated with the resource control. In cases
where the process, task, or project associated with the control possesses fewer
capabilities than allowable by the current value, the value returned by
\fBrctlblk_get_enforced_value()\fR will differ from that returned by
\fBrctlblk_get_value()\fR. This capability difference arises with processes
using an address space model smaller than the maximum address space model
supported by the system.
.sp
.LP
The \fBrctlblk_size()\fR function returns the size of a resource control block
for use in memory allocation. The \fBrctlblk_t *\fR type is an opaque pointer
whose size is not connected with that of the resource control block itself. Use
of \fBrctlblk_size()\fR is illustrated in the example below.
.SH RETURN VALUES
.sp
.LP
The various set routines have no return values. Incorrectly composed resource
control blocks will generate errors when used with \fBsetrctl\fR(2) or
\fBgetrctl\fR(2).
.SH ERRORS
.sp
.LP
No error values are returned. Incorrectly constructed resource control blocks
will be rejected by the system calls.
.SH EXAMPLES
.LP
\fBExample 1 \fRDisplay the contents of a fetched resource control block.
.sp
.LP
The following example displays the contents of a fetched resource control
block.

.sp
.in +2
.nf
#include <rctl.h>
#include <stdio.h>
#include <stdlib.h>

int
main()
{
        rctlblk_t *rblk;
        int rsignal, raction;
rctlblk_t *rblk;
int rsignal;
int raction;

        if ((rblk = malloc(rctlblk_size())) == NULL) {
if ((rblk = malloc(rctlblk_size())) == NULL) {
                (void) perror("rblk malloc");
                exit(1);
        }
}

        if (getrctl("process.max-cpu-time", NULL, rblk, RCTL_FIRST) == -1) {
if (getrctl("process.max-cpu-time", NULL, rblk, RCTL_FIRST) == -1) {
                (void) perror("getrctl");
                exit(1);
        }
}

main()
{
        raction = rctlblk_get_local_action(rblk, &rsignal),
        (void) printf("Resource control for %s\en",
            "process.max-cpu-time");
        (void) printf("Process ID:     %d\en",
            (int)rctlblk_get_recipient_pid(rblk));
        (void) printf("Privilege:      %x\en",
        rctlblk_get_recipient_pid(rblk));
    (void) printf("Privilege:      %x\en"
            rctlblk_get_privilege(rblk));
        (void) printf("Global flags:   %x\en",
    (void) printf("Global flags:   %x\en"
            rctlblk_get_global_flags(rblk));
        (void) printf("Global actions: %x\en",
    (void) printf("Global actions: %x\en"
            rctlblk_get_global_action(rblk));
        (void) printf("Local flags:    %x\en",
    (void) printf("Local flags:    %x\en"
            rctlblk_get_local_flags(rblk));
        (void) printf("Local action:   %x (%d)\en",
    (void) printf("Local action:   %x (%d)\en"
            raction, raction == RCTL_LOCAL_SIGNAL ? rsignal : 0);
        (void) printf("Value:          %llu\en",
            rctlblk_get_value(rblk));
        (void) printf("\tEnforced value: %llu\en",
            rctlblk_get_enforced_value(rblk));

        return (0);
}
.fi
.in -2

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Evolving
_
MT-Level        MT-Safe
.TE
```

```
.SH SEE ALSO
.sp
.LP
\fBrctladm\fR(1M), \fBgetrctl\fR(2), \fBsetrctl\fR(2), \fBgethrtime\fR(3C),
\fBattributes\fR(5)
```