

```

*****
21193 Sun Jan 31 14:08:51 2016
new/usr/src/man/man5/resource_controls.5
6591 resource_controls(5) should talk about project.cpu-cap not project.cpu-caps
*****
1 \" te
2.\" Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
3.\" The contents of this file are subject to the terms of the Common Development
4.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5.\" When distributing Covered Code, include this CDDL HEADER in each file and in
6.TH RESOURCE_CONTROLS 5 "Jul 19, 2013"
7.SH NAME
8 resource_controls \- resource controls available through project database
9.SH DESCRIPTION
10 .sp
10 .LP
11 The resource controls facility is configured through the project database. See
12 \fBproject\fR(4). You can set and modify resource controls through the
13 following utilities:
14 .RS +4
15 .TP
16 .ie t \(\bu
17 .el o
18 \fBprctl\fR(1)
19 .RE
20 .RS +4
21 .TP
22 .ie t \(\bu
23 .el o
24 \fBprojadd\fR(1M)
25 .RE
26 .RS +4
27 .TP
28 .ie t \(\bu
29 .el o
30 \fBprojmod\fR(1M)
31 .RE
32 .RS +4
33 .TP
34 .ie t \(\bu
35 .el o
36 \fBprctladm\fR(1M)
37 .RE
38 .sp
39 .LP
40 In a program, you use \fBsetrctl\fR(2) to set resource control values.
41 .sp
42 .LP
43 In addition to the preceding resource controls, there are resource pools,
44 accessible through the \fBpooladm\fR(1M) and \fBpoolcfg\fR(1M) utilities. In a
45 program, resource pools can be manipulated through the \fBlibpool\fR(3LIB)
46 library.
47 .sp
48 .LP
49 The following are the resource controls are available:
50 .sp
51 .ne 2
52 .na
53 \fB\fbprocess.max-address-space\fR\fR
54 .ad
55 .sp .6
56 .RS 4n
57 Maximum amount of address space, as summed over segment sizes, that is
58 available to this process, expressed as a number of bytes.
59 .RE

```

```

61 .sp
62 .ne 2
63 .na
64 \fB\fbprocess.max-core-size\fR\fR
65 .ad
66 .sp .6
67 .RS 4n
68 Maximum size of a core file created by this process, expressed as a number of
69 bytes.
70 .RE

72 .sp
73 .ne 2
74 .na
75 \fB\fbprocess.max-cpu-time\fR\fR
76 .ad
77 .sp .6
78 .RS 4n
79 Maximum CPU time that is available to this process, expressed as a number of
80 seconds.
81 .RE

83 .sp
84 .ne 2
85 .na
86 \fB\fbprocess.max-data-size\fR\fR
87 .ad
88 .sp .6
89 .RS 4n
90 Maximum heap memory available to this process, expressed as a number of bytes.
91 .RE

93 .sp
94 .ne 2
95 .na
96 \fB\fbprocess.max-file-descriptor\fR\fR
97 .ad
98 .sp .6
99 .RS 4n
100 Maximum file descriptor index available to this process, expressed as an
101 integer.
102 .RE

104 .sp
105 .ne 2
106 .na
107 \fB\fbprocess.max-file-size\fR\fR
108 .ad
109 .sp .6
110 .RS 4n
111 Maximum file offset available for writing by this process, expressed as a
112 number of bytes.
113 .RE

115 .sp
116 .ne 2
117 .na
118 \fB\fbprocess.max-msg-messages\fR\fR
119 .ad
120 .sp .6
121 .RS 4n
122 Maximum number of messages on a message queue (value copied from the resource
123 control at \fBmsgget()\fR time), expressed as an integer.
124 .RE

126 .sp

```

```

127 .ne 2
128 .na
129 \fB\fBprocess.max-msg-qbytes\fR\fR
130 .ad
131 .sp .6
132 .RS 4n
133 Maximum number of bytes of messages on a message queue (value copied from the
134 resource control at \fBmsgget()\fR time), expressed as a number of bytes.
135 .RE

137 .sp
138 .ne 2
139 .na
140 \fB\fBprocess.max-port-events\fR\fR
141 .ad
142 .sp .6
143 .RS 4n
144 Maximum allowable number of events per event port, expressed as an integer.
145 .RE

147 .sp
148 .ne 2
149 .na
150 \fB\fBprocess.max-sem-nsems\fR\fR
151 .ad
152 .sp .6
153 .RS 4n
154 Maximum number of semaphores allowed per semaphore set, expressed as an
155 integer.
156 .RE

158 .sp
159 .ne 2
160 .na
161 \fB\fBprocess.max-sem-ops\fR\fR
162 .ad
163 .sp .6
164 .RS 4n
165 Maximum number of semaphore operations allowed per \fBsemop\fR call (value
166 copied from the resource control at \fBsemget()\fR time). Expressed as an
167 integer, specifying the number of operations.
168 .RE

170 .sp
171 .ne 2
172 .na
173 \fB\fBprocess.max-sigqueue-size\fR\fR
174 .ad
175 .sp .6
176 .RS 4n
177 Maximum number of outstanding queued signals.
178 .RE

180 .sp
181 .ne 2
182 .na
183 \fB\fBprocess.max-stack-size\fR\fR
184 .ad
185 .sp .6
186 .RS 4n
187 Maximum stack memory segment available to this process, expressed as a number
188 of bytes.
189 .RE

191 .sp
192 .ne 2

```

```

193 .na
194 \fB\fBproject.cpu-cap\fR\fR
195 \fB\fBproject.cpu-caps\fR\fR
196 .ad
197 .sp .6
198 .RS 4n
198 Maximum amount of CPU resources that a project can use. The unit used is the
199 percentage of a single CPU that can be used by all user threads in a project.
200 Expressed as an integer. The cap does not apply to threads running in real-time
201 scheduling class. This resource control does not support the \fBsyslog\fR
202 action.
203 .RE

205 .sp
206 .ne 2
207 .na
208 \fB\fBproject.cpu-shares\fR\fR
209 .ad
210 .sp .6
211 .RS 4n
212 Number of CPU shares granted to a project for use with the fair share scheduler
213 (see \fBFS(7)\fR). The unit used is the number of shares (an integer). This
214 resource control does not support the \fBsyslog\fR action.
215 .RE

217 .sp
218 .ne 2
219 .na
220 \fB\fBproject.max-contracts\fR\fR
221 .ad
222 .sp .6
223 .RS 4n
224 Maximum number of contracts allowed in a project, expressed as an integer.
225 .RE

227 .sp
228 .ne 2
229 .na
230 \fB\fBproject.max-crypto-memory\fR\fR
231 .ad
232 .sp .6
233 .RS 4n
234 Maximum amount of kernel memory that can be used for crypto operations.
235 Allocations in the kernel for buffers and session-related structures are
236 charged against this resource control.
237 .RE

239 .sp
240 .ne 2
241 .na
242 \fB\fBproject.max-locked-memory\fR\fR
243 .ad
244 .sp .6
245 .RS 4n
246 Total amount of physical memory locked by device drivers and user processes
247 (including D/ISM), expressed as a number of bytes.
248 .RE

250 .sp
251 .ne 2
252 .na
253 \fB\fBproject.max-lwps\fR\fR
254 .ad
255 .sp .6
256 .RS 4n
257 Maximum number of LWPs simultaneously available to a project, expressed as an

```

```

258 integer.
259 .RE

261 .sp
262 .ne 2
263 .na
264 \fB\fBproject.max-msg-ids\fR\fR
265 .ad
266 .sp .6
267 .RS 4n
268 Maximum number of message queue IDs allowed for a project, expressed as an
269 integer.
270 .RE

272 .sp
273 .ne 2
274 .na
275 \fB\fBproject.max-port-ids\fR\fR
276 .ad
277 .sp .6
278 .RS 4n
279 Maximum allowable number of event ports, expressed as an integer.
280 .RE

282 .sp
283 .ne 2
284 .na
285 \fB\fBproject.max-sem-ids\fR\fR
286 .ad
287 .sp .6
288 .RS 4n
289 Maximum number of semaphore IDs allowed for a project, expressed as an integer.
290 .RE

292 .sp
293 .ne 2
294 .na
295 \fB\fBproject.max-shm-ids\fR\fR
296 .ad
297 .sp .6
298 .RS 4n
299 Maximum number of shared memory IDs allowed for a project, expressed as an
300 integer.
301 .RE

303 .sp
304 .ne 2
305 .na
306 \fB\fBproject.max-shm-memory\fR\fR
307 .ad
308 .sp .6
309 .RS 4n
310 Total amount of shared memory allowed for a project, expressed as a number of
311 bytes.
312 .RE

314 .sp
315 .ne 2
316 .na
317 \fB\fBproject.max-tasks\fR\fR
318 .ad
319 .sp .6
320 .RS 4n
321 Maximum number of tasks allowable in a project, expressed as an integer.
322 .RE

```

```

324 .sp
325 .ne 2
326 .na
327 \fB\fBproject.pool\fR\fR
328 .ad
329 .sp .6
330 .RS 4n
331 Binds a specified resource pool with a project.
332 .RE

334 .sp
335 .ne 2
336 .na
337 \fB\fBrcap.max-rss\fR\fR
338 .ad
339 .sp .6
340 .RS 4n
341 The total amount of physical memory, in bytes, that is available to processes
342 in a project.
343 .RE

345 .sp
346 .ne 2
347 .na
348 \fB\fBtask.max-cpu-time\fR\fR
349 .ad
350 .sp .6
351 .RS 4n
352 Maximum CPU time that is available to this task's processes, expressed as a
353 number of seconds.
354 .RE

356 .sp
357 .ne 2
358 .na
359 \fB\fBtask.max-lwps\fR\fR
360 .ad
361 .sp .6
362 .RS 4n
363 Maximum number of LWPs simultaneously available to this task's processes,
364 expressed as an integer.
365 .RE

367 .sp
368 .LP
369 The following zone-wide resource controls are available:
370 .sp
371 .ne 2
372 .na
373 \fB\fBzone.cpu-cap\fR\fR
374 .ad
375 .sp .6
376 .RS 4n
377 Sets a limit on the amount of CPU time that can be used by a zone. The unit
378 used is the percentage of a single CPU that can be used by all user threads in
379 a zone. Expressed as an integer. When projects within the capped zone have
380 their own caps, the minimum value takes precedence. This resource control does
381 not support the \fBsyslog\fR action.
382 .RE

384 .sp
385 .ne 2
386 .na
387 \fB\fBzone.cpu-shares\fR\fR
388 .ad
389 .sp .6

```

```

390 .RS 4n
391 Sets a limit on the number of fair share scheduler (FSS) CPU shares for a zone.
392 CPU shares are first allocated to the zone, and then further subdivided among
393 projects within the zone as specified in the \fBproject.cpu-shares\fR entries.
394 Expressed as an integer. This resource control does not support the
395 \fBsyslog\fR action.
396 .RE

398 .sp
399 .ne 2
400 .na
401 \fB\fBzone.max-locked-memory\fR\fR
402 .ad
403 .sp .6
404 .RS 4n
405 Total amount of physical locked memory available to a zone.
406 .RE

408 .sp
409 .ne 2
410 .na
411 \fB\fBzone.max-lwps\fR\fR
412 .ad
413 .sp .6
414 .RS 4n
415 Enhances resource isolation by preventing too many LWPs in one zone from
416 affecting other zones. A zone's total LWPs can be further subdivided among
417 projects within the zone within the zone by using \fBproject.max-lwps\fR
418 entries. Expressed as an integer.
419 .RE

421 .sp
422 .ne 2
423 .na
424 \fB\fBzone.max-msg-ids\fR\fR
425 .ad
426 .sp .6
427 .RS 4n
428 Maximum number of message queue IDs allowed for a zone, expressed as an
429 integer.
430 .RE

432 .sp
433 .ne 2
434 .na
435 \fB\fBzone.max-sem-ids\fR\fR
436 .ad
437 .sp .6
438 .RS 4n
439 Maximum number of semaphore IDs allowed for a zone, expressed as an integer.
440 .RE

442 .sp
443 .ne 2
444 .na
445 \fB\fBzone.max-shm-ids\fR\fR
446 .ad
447 .sp .6
448 .RS 4n
449 Maximum number of shared memory IDs allowed for a zone, expressed as an
450 integer.
451 .RE

453 .sp
454 .ne 2
455 .na

```

```

456 \fB\fBzone.max-shm-memory\fR\fR
457 .ad
458 .sp .6
459 .RS 4n
460 Total amount of shared memory allowed for a zone, expressed as a number of
461 bytes.
462 .RE

464 .sp
465 .ne 2
466 .na
467 \fB\fBzone.max-swap\fR\fR
468 .ad
469 .sp .6
470 .RS 4n
471 Total amount of swap that can be consumed by user process address space
472 mappings and \fBtmpfs\fR mounts for this zone.
473 .RE

475 .sp
476 .LP
477 See \fBzones\fR(5).
478 .SS "Units Used in Resource Controls"
480 .sp
479 .LP
480 Resource controls can be expressed as in units of size (bytes), time (seconds),
481 or as a count (integer). These units use the strings specified below.
482 .sp
483 .in +2
484 .nf
485 Category                Res Ctrl   Modifier  Scale
486 Type String
487 -----
488 Size                    bytes      B         1
489                        KB         2^10
490                        MB         2^20
491                        GB         2^30
492                        TB         2^40
493                        PB         2^50
494                        EB         2^60

496 Time                    seconds    s         1
497                        Ks        10^3
498                        Ms        10^6
499                        Gs        10^9
500                        Ts        10^12
501                        Ps        10^15
502                        Es        10^18

504 Count                    integer    none      1
505                        K         10^3
506                        M         10^6
507                        G         10^9
508                        T         10^12
509                        P         10^15
510                        Es        10^18
511 .fi
512 .in -2

514 .sp
515 .LP
516 Scaled values can be used with resource controls. The following example shows a
517 scaled threshold value:
518 .sp
519 .in +2
520 .nf

```

```

521 task.max-lwps=(priv,1K,deny)
522 .fi
523 .in -2

525 .sp
526 .LP
527 In the \fBproject\fR file, the value \fB1K\fR is expanded to \fB1000\fR:
528 .sp
529 .in +2
530 .nf
531 task.max-lwps=(priv,1000,deny)
532 .fi
533 .in -2

535 .sp
536 .LP
537 A second example uses a larger scaled value:
538 .sp
539 .in +2
540 .nf
541 process.max-file-size=(priv,5G,deny)
542 .fi
543 .in -2

545 .sp
546 .LP
547 In the \fBproject\fR file, the value \fB5G\fR is expanded to \fB5368709120\fR:
548 .sp
549 .in +2
550 .nf
551 process.max-file-size=(priv,5368709120,deny)
552 .fi
553 .in -2

555 .sp
556 .LP
557 The preceding examples use the scaling factors specified in the table above.
558 .sp
559 .LP
560 Note that unit modifiers (for example, \fB5G\fR) are accepted by the
561 \fBprctl\fR(1), \fBprojadd\fR(1M), and \fBprojmod\fR(1M) commands. You cannot
562 use unit modifiers in the project database itself.
563 .SS "Resource Control Values and Privilege Levels"
564 .LP
565 A threshold value on a resource control constitutes a point at which local
566 actions can be triggered or global actions, such as logging, can occur.
567 .sp
568 .LP
569 Each threshold value on a resource control must be associated with a privilege
570 level. The privilege level must be one of the following three types:
571 .sp
572 .ne 2
573 .na
574 \fB\basic\fR
575 .ad
576 .sp .6
577 .RS 4n
578 Can be modified by the owner of the calling process.
579 .RE

581 .sp
582 .ne 2
583 .na
584 \fB\bprivileged\fR
585 .ad

```

```

586 .sp .6
587 .RS 4n
588 Can be modified by the current process (requiring \fBsys_resource\fR privilege)
589 or by \fBprctl\fR(1) (requiring \fBproc_owner\fR privilege).
590 .RE

592 .sp
593 .ne 2
594 .na
595 \fB\bssystem\fR
596 .ad
597 .sp .6
598 .RS 4n
599 Fixed for the duration of the operating system instance.
600 .RE

602 .sp
603 .LP
604 A resource control is guaranteed to have one \fB\bssystem\fR value, which is
605 defined by the system, or resource provider. The \fB\bssystem\fR value represents
606 how much of the resource the current implementation of the operating system is
607 capable of providing.
608 .sp
609 .LP
610 Any number of privileged values can be defined, and only one basic value is
611 allowed. Operations that are performed without specifying a privilege value are
612 assigned a basic privilege by default.
613 .sp
614 .LP
615 The privilege level for a resource control value is defined in the privilege
616 field of the resource control block as \fB\brcctl_basic\fR, \fB\brcctl_privileged\fR,
617 or \fB\brcctl_system\fR. See \fB\bsetrctl\fR(2) for more information. You can use
618 the \fBprctl\fR command to modify values that are associated with basic and
619 privileged levels.
620 .sp
621 .LP
622 In specifying the privilege level of \fB\bprivileged\fR, you can use the
623 abbreviation \fB\bpriv\fR. For example:
624 .sp
625 .in +2
626 .nf
627 task.max-lwps=(priv,1K,deny)
628 .fi
629 .in -2

631 .SS "Global and Local Actions on Resource Control Values"
632 .LP
633 There are two categories of actions on resource control values: global and
634 local.
635 .sp
636 .LP
637 Global actions apply to resource control values for every resource control on
638 the system. You can use \fB\brcctladm\fR(1M) to perform the following actions:
639 .RS +4
640 .TP
641 .ie t \(\bu
642 .el o
643 Display the global state of active system resource controls.
644 .RE
645 .RS +4
646 .TP
647 .ie t \(\bu
648 .el o
649 Set global logging actions.
650 .RE

```

```

651 .sp
652 .LP
653 You can disable or enable the global logging action on resource controls. You
654 can set the \fBsyslog\fR action to a specific degree by assigning a severity
655 level, \fBsyslog=\fR\fIlevel\fR. The possible settings for \fIlevel\fR are as
656 follows:
657 .RS +4
658 .TP
659 .ie t \(\bu
660 .el o
661 \fBdebug\fR
662 .RE
663 .RS +4
664 .TP
665 .ie t \(\bu
666 .el o
667 \fBinfo\fR
668 .RE
669 .RS +4
670 .TP
671 .ie t \(\bu
672 .el o
673 \fBnotice\fR
674 .RE
675 .RS +4
676 .TP
677 .ie t \(\bu
678 .el o
679 \fBwarning\fR
680 .RE
681 .RS +4
682 .TP
683 .ie t \(\bu
684 .el o
685 \fBerr\fR
686 .RE
687 .RS +4
688 .TP
689 .ie t \(\bu
690 .el o
691 \fBcrit\fR
692 .RE
693 .RS +4
694 .TP
695 .ie t \(\bu
696 .el o
697 \fBalert\fR
698 .RE
699 .RS +4
700 .TP
701 .ie t \(\bu
702 .el o
703 \fBemerg\fR
704 .RE
705 .sp
706 .LP
707 By default, there is no global logging of resource control violations.
708 .sp
709 .LP
710 Local actions are taken on a process that attempts to exceed the control value.
711 For each threshold value that is placed on a resource control, you can
712 associate one or more actions. There are three types of local actions:
713 \fBnone\fR, \fBdeny\fR, and \fBsignal=\fR. These three actions are used as
714 follows:
715 .sp
716 .ne 2

```

```

717 .na
718 \fBnone\fR
719 .ad
720 .sp .6
721 .RS 4n
722 No action is taken on resource requests for an amount that is greater than the
723 threshold. This action is useful for monitoring resource usage without
724 affecting the progress of applications. You can also enable a global message
725 that displays when the resource control is exceeded, while, at the same time,
726 the process exceeding the threshold is not affected.
727 .RE

729 .sp
730 .ne 2
731 .na
732 \fBdeny\fR
733 .ad
734 .sp .6
735 .RS 4n
736 You can deny resource requests for an amount that is greater than the
737 threshold. For example, a \fBtask.max-lwps\fR resource control with action deny
738 causes a \fBfork()\fR system call to fail if the new process would exceed the
739 control value. See the \fBfork\fR(2).
740 .RE

742 .sp
743 .ne 2
744 .na
745 \fBsignal=\fR
746 .ad
747 .sp .6
748 .RS 4n
749 You can enable a global signal message action when the resource control is
750 exceeded. A signal is sent to the process when the threshold value is exceeded.
751 Additional signals are not sent if the process consumes additional resources.
752 Available signals are listed below.
753 .RE

755 .sp
756 .LP
757 Not all of the actions can be applied to every resource control. For example, a
758 process cannot exceed the number of CPU shares assigned to the project of which
759 it is a member. Therefore, a deny action is not allowed on the
760 \fBproject.cpu-shares\fR resource control.
761 .sp
762 .LP
763 Due to implementation restrictions, the global properties of each control can
764 restrict the range of available actions that can be set on the threshold value.
765 (See \fBrcctladm\fR(1M).) A list of available signal actions is presented in the
766 following list. For additional information about signals, see
767 \fBsignal\fR(3HEAD).
768 .sp
769 .LP
770 The following are the signals available to resource control values:
771 .sp
772 .ne 2
773 .na
774 \fBSIGABRT\fR
775 .ad
776 .sp .6
777 .RS 4n
778 Terminate the process.
779 .RE

781 .sp
782 .ne 2

```

```

783 .na
784 \fB\fBSIGHUP\fR\fR
785 .ad
786 .sp .6
787 .RS 4n
788 Send a hangup signal. Occurs when carrier drops on an open line. Signal sent to
789 the process group that controls the terminal.
790 .RE

792 .sp
793 .ne 2
794 .na
795 \fB\fBSIGTERM\fR\fR
796 .ad
797 .sp .6
798 .RS 4n
799 Terminate the process. Termination signal sent by software.
800 .RE

802 .sp
803 .ne 2
804 .na
805 \fB\fBSIGKILL\fR\fR
806 .ad
807 .sp .6
808 .RS 4n
809 Terminate the process and kill the program.
810 .RE

812 .sp
813 .ne 2
814 .na
815 \fB\fBSIGSTOP\fR\fR
816 .ad
817 .sp .6
818 .RS 4n
819 Stop the process. Job control signal.
820 .RE

822 .sp
823 .ne 2
824 .na
825 \fB\fBSIGXRES\fR\fR
826 .ad
827 .sp .6
828 .RS 4n
829 Resource control limit exceeded. Generated by resource control facility.
830 .RE

832 .sp
833 .ne 2
834 .na
835 \fB\fBSIGXFSZ\fR\fR
836 .ad
837 .sp .6
838 .RS 4n
839 Terminate the process. File size limit exceeded. Available only to resource
840 controls with the \fB\RCTL_GLOBAL_FILE_SIZE\fR property
841 (\fB\process.max-file-size\fR). See \fB\brctlblk_set_value\fR(3C).
842 .RE

844 .sp
845 .ne 2
846 .na
847 \fB\fBSIGXCPU\fR\fR
848 .ad

```

```

849 .sp .6
850 .RS 4n
851 Terminate the process. CPU time limit exceeded. Available only to resource
852 controls with the \fB\RCTL_GLOBAL_CPU_TIME\fR property
853 (\fB\process.max-cpu-time\fR). See \fB\brctlblk_set_value\fR(3C).
854 .RE

856 .SS "Resource Control Flags and Properties"
861 .sp
867 .LP
858 Each resource control on the system has a certain set of associated properties.
859 This set of properties is defined as a set of flags, which are associated with
860 all controlled instances of that resource. Global flags cannot be modified, but
861 the flags can be retrieved by using either \fB\brctladm\fR(1M) or the
862 \fB\bsetrctl\fR(2) system call.
863 .sp
864 .LP
865 Local flags define the default behavior and configuration for a specific
866 threshold value of that resource control on a specific process or process
867 collective. The local flags for one threshold value do not affect the behavior
868 of other defined threshold values for the same resource control. However, the
869 global flags affect the behavior for every value associated with a particular
870 control. Local flags can be modified, within the constraints supplied by their
871 corresponding global flags, by the \fB\brctl\fR command or the \fB\bsetrctl\fR
872 system call. See \fB\bsetrctl\fR(2).
873 .sp
874 .LP
875 For the complete list of local flags, global flags, and their definitions, see
876 \fB\brctlblk_set_value\fR(3C).
877 .sp
878 .LP
879 To determine system behavior when a threshold value for a particular resource
880 control is reached, use \fB\brctladm\fR to display the global flags for the
881 resource control. For example, to display the values for
882 \fB\process.max-cpu-time\fR, enter:
883 .sp
884 .in +2
885 .nf
886 $ rctladm process.max-cpu-time
887 process.max-cpu-time  syslog=off [ lowerable no-deny cpu-time inf seconds ]
888 .fi
889 .in -2

891 .sp
892 .LP
893 The global flags indicate the following:
894 .sp
895 .ne 2
896 .na
897 \fB\blowerable\fR\fR
898 .ad
899 .sp .6
900 .RS 4n
901 Superuser privileges are not required to lower the privileged values for this
902 control.
903 .RE

905 .sp
906 .ne 2
907 .na
908 \fB\bnodeny\fR\fR
909 .ad
910 .sp .6
911 .RS 4n
912 Even when threshold values are exceeded, access to the resource is never
913 denied.

```

```

914 .RE
916 .sp
917 .ne 2
918 .na
919 \fB\fBcpu-time\fR\fR
920 .ad
921 .sp .6
922 .RS 4n
923 \fBSIGXCPU\fR is available to be sent when threshold values of this resource
924 are reached.
925 .RE

927 .sp
928 .ne 2
929 .na
930 \fB\fBseconds\fR\fR
931 .ad
932 .sp .6
933 .RS 4n
934 The time value for the resource control.
935 .RE

937 .sp
938 .LP
939 Use the \fBprctl\fR command to display local values and actions for the
940 resource control. For example:
941 .sp
942 .in +2
943 .nf
944 $ prctl -n process.max-cpu-time $$
945   process 353939: -ksh
946   NAME      PRIVILEGE  VALUE      FLAG  ACTION  RECIPIENT
947   process.max-cpu-time
948   privileged 18.4Es    inf    signal=XCPU  -
949   system    18.4Es    inf    none
950 .fi
951 .in -2

953 .sp
954 .LP
955 The \fBmax\fR (\fBRTL_LOCAL_MAXIMAL\fR) flag is set for both threshold values,
956 and the \fBinf\fR (\fBRTL_GLOBAL_INFINITE\fR) flag is defined for this
957 resource control. An \fBinf\fR value has an infinite quantity. The value is
958 never enforced. Hence, as configured, both threshold quantities represent
959 infinite values that are never exceeded.
960 .SS "Resource Control Enforcement"
961 .sp
962 More than one resource control can exist on a resource. A resource control can
963 exist at each containment level in the process model. If resource controls are
964 active on the same resource at different container levels, the smallest
965 container's control is enforced first. Thus, action is taken on
966 \fBprocess.max-cpu-time\fR before \fBtask.max-cpu-time\fR if both controls are
967 encountered simultaneously.
968 .SH ATTRIBUTES
969 .sp
970 See \fBattributes\fR(5) for a description of the following attributes:
971 .sp

973 .sp
974 .TS
975 box;
976 c | c
977 1 | 1 .

```

```

978 ATTRIBUTE TYPE  ATTRIBUTE VALUE
979 _
980 Interface Stability  Evolving
981 .TE

983 .SH SEE ALSO
984 .LP
985 \fBprctl\fR(1), \fBpooladm\fR(1M), \fBpoolcfg\fR(1M), \fBprojadd\fR(1M),
986 \fBprojmod\fR(1M), \fBbrctladm\fR(1M), \fBsetrctl\fR(2),
987 \fBbrctlblk_set_value\fR(3C), \fBlibpool\fR(3LIB), \fBproject\fR(4),
988 \fBattributes\fR(5), \fBfss\fR(7)
989 .sp
990 .LP
991 \fISystem Administration Guide: Virtualization Using the Solaris Operating
992 System\fR

```