JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

VDCF - Virtual Datacenter Control Framework

for the Solaris™ Operating System

Resource Management

Version 3.0
3. September 2010

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

Contents

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

1 Introduction

This documentation describes the Resource Management (RM) feature of the Virtual Datacenter Control Framework (VDCF) for the Solaris Operating System and explains how to use this feature.

This feature is an extension to the VDCF vServer product.

See these documents for more information about the related VDCF products:

VDCF Base – Administration Guide	for information about VDCF base product usage
VDCF vServer – Administration Guide	for information about VDCF vServer product usage
VDCF – Monitoring	for information about VDCF Monitoring usage

1.1 Overview

VDCF Resource Management is a VDCF Enterprise extension available to VDCF Enterprise customers.

VDCF Resource Management is based on Solaris 10 Resource Manager. It delivers central control of resources that can be assigned to a particular Zone. Extended Zone Resource Management has been integrated into Solaris 10 Update 4 (8/07). With Solaris 10 Update 5 (5/08) the Resource Manager got a new resource control 'cpu-cap'.

The basic functionality of VDCF Resource Management relies on this Solaris feature set.

While VDCF Resource Management is used to configure resource limits, VDCF Monitoring offers features to collect and display the resource usage.

1.1.1 CPU Resource Management

Solaris Resource Manager delivers two distinct technologies to control the amount of CPU being used by a Zone or as we call it in VDCF terminology: vServer. One of these technologies are Dynamic Resource Pools. Such a pool can be configured to contain a number of CPU's as seen by the Solaris scheduler. A pool can be configured with a static number or a range of CPU's. If a range has been specified, the number of CPU's can and will vary based on the defined importance for that pool. As of Solaris 10 8/07 the pool can be configured using the Zone configuration. As soon as a Zone boot up, the pool gets created automatically. The Zone can only boot up if sufficient CPU's are available. Note that a CPU can not be shared between different pools. The granularity for the assignment of CPU's to vServer therefor is 1 CPU.

The other way to control the amount of CPU being used by a Zone is the Fair Share Scheduler (FSS). It allows for fine granular assignment of CPU resources in terms of shares. FSS allows for sub CPU granularity.

To fully support the approach of utility based computing, we need to base the CPU Resource Management on a performance metric. This allows an administrator to set up a vServer with a particular performance in terms of CPU power. The metric used to specify the power of a particular vServer, should be generic and automatically translated into the number of CPU shares. This would allow the migration of vServer from one node to another, even if this new node has more or less total CPU power available. A migration of a vServer requires an adaption of its assigned shares so, that its available CPU power remains the same.

VDCF Resource Management supports both technologies. However, true utility based computing can only be achieved using the FSS based mechanism.

1.1.2 Memory Resource Management

Memory can be controlled on different levels using the Solaris Resource Manager. As of Solaris 10 8/07 it is possible to control the amount of physical memory (RAM), the amount of virtual memory (SWAP), the amount of shared memory and the amount of locked down memory allowed for a particular vServer.

VDCF Resource Management support all mechanisms for controlling resources related to a Solaris Zone. More information about the Solaris 10 Resource Manager and its abilities can be found on http://docs.sun.com/app/docs/doc/817-1592.

1.2 Requirements / Patches

The VDCF Resource Management implementation is based on Solaris 10 8/07 (Update 4) features. To use this feature the target Nodes must run Solaris 10 8/07 or later. It is supported to use an older Solaris 10 Release (Update1,2,3) with Kernel Patch 120011-14 (sparc) or 120012-14 (i386) or later.

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

2 Installation and Configuration

2.1 Prerequisites

The JSvdcf-rm package requires the following VDCF packages to be installed on the VDCF Management Server:

- JSvdcf-base 3.0.0 or later
- JSvdcf-vserver 3.0.0 or later

2.2 Installation

a) sparc platform

```
cd <download-dir>
pkgadd -d ./JSvdcf-rm_<version>_sparc.pkg
```

b) i386 platform

```
cd <download-dir>
pkgadd -d ./JSvdcf-rm_<version>_i386.pkg
```

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

2.3 Configuration

2.3.1 Granting User Access

The VDCF Resource Management package introduces the RBAC Profile “VDCF resource Module”. Assign this RBAC profile to your admin users.

2.3.2 Base Configuration

VDCF Resource Management (RM) uses a database table to define the relative performance of a specific CPU type. This performance metric will be used to define a servers relative CPU power. To list all defined CPU's and there performance metrics use the following command:

```
# rcadm -c show_perf cpu
```

```
        Performance Base Units set to: 100

            CPU-Model    CPU-Freq      Perf-Value    Base-Units
         UltraSPARC-IV+    2100          617           6.17
         UltraSPARC-IV+    1800          570           5.7
         UltraSPARC-IV+    1950          554           5.54
         UltraSPARC-IV+    1500          502           5.02
             SPARC64-VI    2400          400           4
        UltraSPARC-IIIi    1600          389           3.89
        UltraSPARC-IIIi    1593          387           3.87
        UltraSPARC-IIIi    1504          378           3.78
        UltraSPARC-IIIi    1500          377           3.77
        UltraSPARC-IIIi    1503          377           3.77
             SPARC64-VI    2280          360           3.6
        UltraSPARC-III+    1350          360           3.6
             SPARC64-VI    2150          352           3.52
        UltraSPARC-IIIi    1336          342           3.42
          UltraSPARC-IV    1350          337           3.37
        UltraSPARC-IIIi    1280          329           3.29
        UltraSPARC-III+    1200          323           3.23
          UltraSPARC-IV    1200          315           3.15
        UltraSPARC-III+    1050          301           3.01
          UltraSPARC-IV    1050          293           2.93
        UltraSPARC-III+    900           269           2.69
        UltraSPARC-IIIi    1002          269           2.69
        UltraSPARC-III+    750           228           2.28
         UltraSPARC-IIe    650           222           2.22
         UltraSPARC-IIe    548           189           1.89
         UltraSPARC-IIe    500           176           1.76
         UltraSPARC-IIe    502           175           1.75
          UltraSPARC-II    480           171           1.71
          UltraSPARC-II    450           165           1.65
```

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

```
    SPARC64-IV    603          150          1.5
 UltraSPARC-T2    1417         142          1.42
 UltraSPARC-II    400          136          1.36
 UltraSPARC-T1    1400         135          1.35
UltraSPARC-T2+    1415         128          1.28
 UltraSPARC-T2    1167         124          1.24
 UltraSPARC-T1    1200         118          1.18
UltraSPARC-T2+    1162         112          1.12
 UltraSPARC-II    296          111          1.11
 UltraSPARC-T1    1000         100          1
```

This command lists all the known SPARC CPU types available on the market and the frequencies. Assigned to each CPU there is a performance metric that indicates the relative performance of that particular CPU (third column). The fourth column indicates the performance of a CPU type in so called Base Unit values. Base Units are used to define a vServers CPU power.

As indicated in the first line of the output, the Base Units are set to 100 in this example. This default value can be modified by adding another value in `conf/customize.cfg`.

```
#
# PERFORMANCE METRIC BASE SETTING
export PERF_BASE_UNIT="10"
```

It allows a customer to set its preferred CPU type to be represented as 1 Base Unit. In this example, an `UltraSPARC-T1` CPU (as represented as a dispatchable unit in Solaris) is defined as 1 Base Unit. To define the relative CPU power of a particular system, this value will be multiplied by the number of CPU's visible by the OS. Discovering a new physical server using the `nodecfg` command will automatically set up the appropriate metrics for it. To list the defined values use:

```
# rcadm -c show_perf node

   Name ... CPU-Model           CPU-Freq  CPUs   Mem MB    Base-Units
  s0003 ... UltraSPARC-IIIi     1002      2      2048      11.1
  s0004 ... UltraSPARC-IIIi     1280      2      2048      14.08
  s0005 ... UltraSPARC-T1       1000      24     8184      24
```

To stick with the above example, the output for the `UltraSPARC-T1` shows a total computing power of 24 Base Units. These Base Units can now be allocated to vServers. It guarantees the requested CPU power in case multiple vServers are contending for CPU resources on that system. However, it is not possible to cap CPU consumption on that defined value. This is a Solaris 10 Resource Manager rather then a VDCF RM restriction. The implementation of CPU capping will be available in a future Solaris 10 Update. As soon as this feature will be made available, VDCF RM will take advantage of it.

2.3.1 Resource Rules

New since VDCF Resource Management 3.0

To avoid assigning too much memory to vServers, VDCF Resource Management allows to define a Minimum Memory required for the Node.

Default value:

```
# RESOURCE RULES
# - Minimum RAM required/reserved for NODE in %
export RESOURCE_NODE_RAM_MIN=10
```

Defining too much Memory to a vServer is rejected by the rcadm set operation. This check may be disabled by using the '`force`' flag.

```
s0002:~$ rcadm -c set vserver=s0152 ram=1024
setting rctl properties for vServer s0152
ERROR: Not enough Minimum Memory (192 MB / 10%) for Node s0006, Available 704 MB
ERROR: failed to set rctl properties
```

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

3 Usage

The whole concept of VDCF RM is based on the idea that vServers are used to consolidate applications. This consolidation uses a pool of compute resources where a server is called a node. vServers are hosted on that pool with great flexibility. A vServer can be migrated from one physical node to another. VDCF takes care of all the required operations. VDCF RM can be used to ensure that a particular vServer gets enough resources while it is running in parallel with other vServers on the same node.

VDCF RM uses the `rcadm` command to control resource assignments centrally for all vServers managed by VDCF.

3.1 Available Resource Controls

`rcadm` manages Solaris 10 Resource Manager Resource Controls as properties. CPU Resources are either controlled by Fair Share Scheduler (FSS) or by Dynamic Resource Pool (DRP) feature. For information about this two technologies see Chapter 1.1. Overview. The two technologies are mutually exclusive. That is, one either specifies CPU resources by FSS or DRP.

The following properties are available to activate FSS based CPU control:

'CPU_Shares'	Number of Base Units. If CPU_Shares are defined 'CPUs' and 'Importance' are not allowed and FSS based RM is activated
'CPU_cap'	CPU capping in Base Units. 'CPU_Shares' and 'CPU_cap' can be but must not be specified together. They indicate a guaranteed and a maximum CPU entitlement.

If DRP should be activated the following properties needs to be set:

'CPUs'	Number of CPU's for vServer. Can be a range.
'Importance'	Relative importance of vServer.

VDCF RM also supports the management of all other supported zone resource controls. These are:

'RAM'	Physical RAM in K,M,G,T
'SWAP'	Virtual Memory in K,M,G,T
'Locked'	Maximum locked down Memory in K,M,G,T
'LWP'	Maximum number of LWPs
'MSG_ids'	Maximum number of Message Queues
'SEM_ids'	Maximum number of Semaphores
'SHM_ids'	Maximum number of Shared Memory Segments
'SHM_Size'	Maximum size of all Shared Memory Segments in K,M,G,T

Sizes are specified as n, n[bB], n[kK], n[mM], n[gG], n[tT] where n is megabytes. The minimum values are: 1048576b, 1024k, 1 or 1m, 1g, 1t.

Properties are not case sensitive!

For more information about the `rcadm` command use `rcadm -H`.

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

3.2 Basic vServer Operation

Properties are either set or unset for a particular vServer. Additionally, each property, set or unset, is either current or uncommitted. Current means that they are currently activated for the vServer. Uncommitted properties are defined but not activated for a vServer. However, the next `commit` operation will activate those settings. Uncommitted settings might be abandoned using the `revert` operation.

To check property settings and state for a vServer called s1022 use the following command:

```
# rcadm -c show vserver=s1022

       Resource Management settings for: s1022

  Property Name   Current Setting       Uncommitted Setting   Units
     CPU_Shares:  6                     -                     BaseUnits
           CPUs:  -                     -                     CPUs
     Importance:  -                     -                     Weight
            RAM:  1024                  -                     MB
           SWAP:  2048                  -                     MB
         Locked:  512                   -                     MB
            LWP:  -                     -                     Units
        MSG_ids:  -                     -                     Units
        SEM_ids:  10                    -                     Units
        SHM_ids:  10                    -                     Units
       SHM_Size:  800                   -                     MB
```

Various resource controls have been set and activated for that vServer. There are currently no uncommitted settings for it. To change a current, activate a new or abandon a existing setting use the following commands:

```
# rcadm -c set   vserver=s1022 lwp=1000 cpu_shares=4
# rcadm -c unset vserver=s1022 sem_ids
```

Check your changes with:

```
# rcadm -c show vserver=s1022

       Resource Management settings for: s1022

  Property Name   Current Setting       Uncommitted Setting   Units
     CPU_Shares:  6                     4                     BaseUnits
           CPUs:  -                     -                     CPUs
     Importance:  -                     -                     Weight
            RAM:  1024                  1024                  MB
           SWAP:  2048                  2048                  MB
         Locked:  512                   512                   MB
            LWP:  -                     1000                  Units
        MSG_ids:  -                     -                     Units
        SEM_ids:  10                    -                     Units
        SHM_ids:  10                    10                    Units
       SHM_Size:  800                   800                   MB
```

The requested changes are now stored in the database but no changes have been made on the actual running configuration for the vServer. To activate those changes, submit the following command:

```
# rcadm -c commit vserver=s1022
```

If one prefers not to implement the changes, the revert operation can be used:

```
# rcadm -c revert vserver=s1022
```

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

3.3 Observing Node Settings

Once vServers have been configured to be resource controlled, it might be interesting to see what the impact on the most constrained resources are. To generate an overview list of all resource managed vServer use the following command:

```
# rcadm -c show

vServer  RCTL     ---- Base  Units ----       Memory   Node    vServer    vServer
Name     State    Total      CPU Shares  CPUs Size MB  Name    CPU Usage  Mem Usage
s1022    ACTIVE   11.1        6          -     1024    s0003   54.05%     50.00%
s1023    ACTIVE   11.1        3          -      512    s0003   27.02%     25.00%
s1024    ACTIVE   20.0       16          -     4096    s0005   80.00%     75.00%
```

To see what a particular Node has in terms of resource management settings and what its current occupation is, issue the following command:

```
# rcadm -c show name=s0003
```

The `name` argument takes either a Node or a vServer name. The output triggered by the `name` argument, always lists the respective Node and its assigned vServers.

```
   Node  --- Base Units ---      Total      Total      Total
   Name  Total       Used       Memory MB  CPU Usage  Mem Usage
  s0003  11.1        9          2048        81.07%     75.00%

vServer  RCTL       Base Units  Used       Memory     vServer    vServer
   Name  State      CPU Shares  CPUs       Size MB    CPU Usage  Mem Usage
  s1022  ACTIVE     6           -          1024        54.05%     50.00%
  s1023  ACTIVE     3           -           512        27.02%     25.00%
```

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

3.4 Customization of the VDCF environment

These VDCF configuration values can be changed to adjust VDCF for a customer environment. To overwrite a VDCF variable add the appropriate value to `customize.cfg`:

Variable name	Description
`RESOURCE_NODE_RAM_MIN`	% of physical RAM reserved for the global zone. Defaults to 10%. You can't assign more than 90% of the RAM to your vServers.

4 Manpages

```
User Commands                                rcadm -c <operation>(1)


NAME
     rcadm -c  <operation>  -  Command  for  Resource  Management
     operations


SYNOPSIS
     rcadm -c <operation>


DESCRIPTION
     Command to show and manipulate resource control settings for
     vServers.


        rcadm -c show

        rcadm -c show_perf

        rcadm -c statistics

        rcadm -c set

        rcadm -c unset

        rcadm -c commit

        rcadm -c clone

        rcadm -c revert

        rcadm -c remove



     Use the rcadm command  without  arguments  to  get  detailed
     information about available resource control properties.



     Use the rcadm -H <operation> command to get detailed  infor-
     mation about how to use a particular operation.


NOTE
     The rcadm framework requires at least Solaris 10 8/07.


RETURN VALUES
     rcadm returns 0 if the operation was successful or not equal
     0 if failed.



JomaSoft             Last change: 2008 Jun 2                  1
```

User Commands rcadm -c <operation>(1)

SEE ALSO
rcadm_show(1M), rcadm_show_perf(1M), rcadm_statistics(1M), rcadm_set(1M), rcadm_unset(1M), rcadm_clone(1M), rcadm_revert(1M), rcadm_remove(1M), rcadm_commit(1M)

NOTES

JomaSoft Last change: 2008 Jun 2 2

```
User Commands                                        rcadm -c clone(1)


NAME
     rcadm -c clone - Used to clone resource control settings  of
     a particular vServer


SYNOPSIS
     rcadm -c clone
     vserver=<vserver list>
     template=<vserver>
     [ force ]


DESCRIPTION
     rcadm -c clone is used to clone or copy all resource control
     settings  of  a vServer specified in 'template'. A new entry
     for 'vserver' will  be  created  and  populated  with  these
     values.  This  operation will be kept local in a uncommitted
     state. To transfer the newly created settings to the vServer
     specified  in  'vserver' use the rcadm -c commit command. To
     inspect the actual and pending settings for  a  vServer  use
     the rcadm -c show command.

     The force flag is used to disable resource rules checking.


EXAMPLES
     Following an example for rcadm -c clone

        rcadm -c clone vserver=s1025 template=s1024


RETURN VALUES
     rcadm -c clone returns 0 if the operation was successful  or
     not equal 0 if failed.


SEE ALSO
     rcadm_show(1M),  rcadm_show_perf(1M),  rcadm_statistics(1M),
     rcadm_set(1M),       rcadm_unset(1M),      rcadm_revert(1M),
     rcadm_remove(1M), rcadm_commit(1M)




JomaSoft            Last change: 2010 Sep 3                  1
```

```
User Commands                                     rcadm -c commit(1)


NAME
     rcadm -c commit - Used to commit settings


SYNOPSIS
     rcadm -c commit
     vserver=<vserver list>
     [ push ]
     [ force ]


DESCRIPTION
     rcadm -c commit is used to  commit  pending  settings  to  a
     vServer.  This operation applies all uncommitted settings to
     the running vServer. If no uncommitted settings  are  avail-
     able,  nothing  happens.  To  inspect the actual and pending
     settings for a vServer use the rcadm  -c  show  command.  To
     re-apply  current (not pending or uncommitted) settings to a
     vServer, use the 'push' flag. To transfer  pending  settings
     to  a  not running node or detached vServer, use the 'force'
     flag. The settings will then be  reapplied  if  the  vServer
     attaches  to  its  node. In case where the node was not run-
     ning, the settings have note been  stored  for  the  vServer
     using  'force'.  Use 'push' to apply settings after the node
     has been restarted.


EXAMPLES
     Following an example for rcadm -c commit

        rcadm -c commit vserver=s1020


NOTE
     The rcadm framework requires at least Solaris 10 8/07.


RETURN VALUES
     rcadm -c commit returns 0 if the operation was successful or
     not equal 0 if failed.


SEE ALSO
     rcadm_show(1M),  rcadm_show_perf(1M),  rcadm_statistics(1M),
     rcadm_set(1M),       rcadm_unset(1M),       rcadm_clone(1M),
     rcadm_revert(1M), rcadm_remove(1M), rcadm_commit(1M)




JomaSoft            Last change: 2007 Sep 20                  1
```

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

NAME

rcadm -c remove - Used to remove resource control settings for a vServer entirely

SYNOPSIS

```
rcadm -c remove
vserver=<vserver list>
```

DESCRIPTION

rcadm -c remove is used to remove all resource control property settings from a vServer. This operations are always kept local in a uncommitted state. To remove the settings on a particular vServer use the rcadm -c commit command. To inspect the actual and pending settings for a vServer use the rcadm -c show command.

EXAMPLES

Following an example for rcadm -c remove

```
rcadm -c remove vserver=s1020,s1024
```

RETURN VALUES

rcadm -c remove returns 0 if the operation was successful or not equal 0 if failed.

SEE ALSO

rcadm_show(1M), rcadm_show_perf(1M), rcadm_statistics(1M), rcadm_set(1M), rcadm_unset(1M), rcadm_clone(1M), rcadm_revert(1M), rcadm_remove(1M), rcadm_commit(1M)

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

NAME

rcadm -c revert - Used to revert uncommitted settings

SYNOPSIS

```
rcadm -c revert
vserver=<vserver list>
```

DESCRIPTION

rcadm -c revert is used to get rid of any uncommitted settings. All operations are kept local. No actions are performed on the target vServer. To inspect the actual and pending settings for a vServer use the rcadm -c show command.

EXAMPLES

Following an example for rcadm -c revert

```
rcadm -c revert vserver=s1024
```

RETURN VALUES

rcadm -c revert returns 0 if the operation was successful or not equal 0 if failed.

SEE ALSO

rcadm_show(1M), rcadm_show_perf(1M), rcadm_statistics(1M), rcadm_set(1M), rcadm_unset(1M), rcadm_clone(1M), rcadm_revert(1M), rcadm_remove(1M), rcadm_commit(1M)

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

NAME

rcadm -c set - Used to set resource control properties

SYNOPSIS

```
rcadm -c set help

rcadm -c set
vserver=<vserver list>
<property>=<property value> ...
[ force ]
```

DESCRIPTION

rcadm -c set is used to set various resource control properties on a vServer. The first form of the command specifying the 'help' flag shows a more detailed help explaining the valid properties.

If no properties exist while the set command is run, they will be created for the vServer specified in 'vserver'. Settings created or changed with this command, are always kept local in a uncommitted state. To transfer the settings to the particular vServer use the rcadm -c commit command. To inspect the actual and pending settings for a vServer use the rcadm -c show command.

The force flag is used to disable resource rules checking. Current rules checking is based on configuration of RESOURCE_NODE_RAM_MIN.

Valid properties are defined as a fixed list. These Properties translate into Solaris 10 Resource Controls (see prctl(1M)). Following a list of all defined properties and their meanings:

CPU_Shares/CPU_cap and CPUs/Importance Properties are mutually exclusive

'CPU_Shares'
Number of Base Units. If CPU_Shares are defined 'CPUs' and 'Importance' are not allowed and FSS based RM is activated

'CPU_cap'
CPU capping in Base Units. 'CPU_Shares' and 'CPU_cap' can be but must not be specified together. They indicate a guaranteed and a maximum CPU entitlement.

```
  -- or --
     'CPUs'
          Number of CPUs. Requires 'Importance'.

     'Importance'
          Relative importance of temp pool


  General Properties with no additional dependencies
     'RAM'
          Physical RAM in K,M,G,T

     'SWAP'
          Virtual Memory in K,M,G,T

     'Locked'
          Maximum locked down Memory in K,M,G,T

     'LWP'
          Maximum number of LWPs

     'MSG_ids'
          Maximum number of Message Queues

     'SEM_ids'
          Maximum number of Semaphores

     'SHM_ids'
          Maximum number of Shared Memory Segments

     'SHM_Size'
          Maximum size of all Shared Memory Segments in K,M,G,T

          Sizes are specified as n, n[bB], n[kK],  n[mM],  n[gG],
          n[tT]  where  n  is  megabytes. The minimum values are:
          1048576b, 1024k, 1 or 1m, 1g, 1t.

          Properties are not case sensitive!


EXAMPLES
     Following an example for rcadm -c set command.

        rcadm -c set vserver=s1024 cpu_shares=2 ram=2g


RETURN VALUES
     rcadm -c set returns 0 if the operation  was  successful  or
     not equal 0 if failed.
```

SEE ALSO
rcadm_show(1M), rcadm_show_perf(1M), rcadm_statistics(1M), rcadm_unset(1M), rcadm_clone(1M), rcadm_revert(1M), rcadm_remove(1M), rcadm_commit(1M)

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

NAME
rcadm -c show - Shows resource control settings

SYNOPSIS
rcadm -c show
[name=<node or vserver name>]
[vserver=<vserver name>]
[all]

DESCRIPTION
rcadm -c show invoked with no additional arguments, shows the current resource control settings for vServers. Only those vServer with resource control settings attached to them are shown. But when the vPool feature is activated only vServers assigned to the users vPools are listed. In that case the 'all' flag may be used to list all existing vServers with resource control settings (even those assigned to other users).

Using the 'vserver' argument with the show command , lists the current and if available, also the uncommitted settings for this particular vServer.

The 'name' argument is used to list the relation of a vServer to its hosting node. 'name' accepts either a node or a vServer name and lists resource management information for the node and all its currently assigned vServers.

EXAMPLES
Following an example for rcadm -c show

```
rcadm -c show vserver=s1020
```

RETURN VALUES
rcadm -c show returns 0 if the operation was successful or not equal 0 if failed.

SEE ALSO
rcadm_show(1M), rcadm_show_perf(1M), rcadm_statistics(1M), rcadm_set(1M), rcadm_unset(1M), rcadm_clone(1M), rcadm_revert(1M), rcadm_remove(1M), rcadm_commit(1M)

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

NAME

rcadm -c show_perf - Lists the the node or cpu performance values

SYNOPSIS

```
rcadm -c show_perf
{ node | cpu }
```

DESCRIPTION

rcadm -c show_perf cpu lists all defined cpu types, frequencies, performance values and their assigned base units. cpu type and frequency is used to uniquely identify a particular processor. The relative performance of a processor is defined by its Perf-Value. These values can be derived from SPECint or similar estimates and represent relative performance of a particular processor. In the rcadm framework, one does not directly deal with these relative performance values. Instead, a so called base unit has been invented. A base unit can be set to any useful number using the PERF_BASE_UNIT property in the customize.cfg configuration file. It is used to express computing power for a vServer in a normalized way (see EXAMPLES).

rcadm -c show_perf node extrapolates the above discussed values for all nodes registered in this compute pool. It shows the relevant performance numbers for all this nodes.

EXAMPLES

Following an example for rcadm -c show_perf cpu:

The sample output below shows the values for various processor types normalized on a single T1 thread running at 1GHz.

```
rcadm -c show_perf cpu

     Performance Base Units set to: 100

       CPU-Model     CPU-Freq       Perf-Value      Base-
Units
  UltraSPARC-IV+    1500          800           8
  UltraSPARC-IV+    1800          960           9.6
   UltraSPARC-IV    1200          600           6
   UltraSPARC-IV    1350          675           6.75
 UltraSPARC-IIIi    1002          555           5.55
 UltraSPARC-IIIi    1280          704           7.04
 UltraSPARC-IIIi    1336          706           7.06
 UltraSPARC-IIIi    1503          794           7.94
 UltraSPARC-IIIi    1504          794           7.94
 UltraSPARC-IIIi    1593          845           8.45
```

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

UltraSPARC-III+	900	533	5.33
UltraSPARC-III+	1050	622	6.22
UltraSPARC-III+	1200	711	7.11
UltraSPARC-III+	1350	800	8
UltraSPARC-T1	1000	100	1
UltraSPARC-T1	1200	120	1.2
UltraSPARC-T1	1400	140	1.4
UltraSPARC-II	450	200	2
UltraSPARC-IIe	548	174	1.74

Following an example for rcadm -c show_perf node:

```
rcadm -c show_perf node

  Name  Model   CPU-Model CPUs  Mem MB  Base-Units
 s0003  V240    US-IIIi    2     2048      11.1
 s0004  V240    US-IIIi    2     2048      14.08
 s0005  T200    US-T1     24     8184      24
 s0006  V240    US-IIIi    2     2048      14.08
 s0008  V240    US-IIIi    2     2048      14.08
```

RETURN VALUES

rcadm -c show_perf returns 0 if the operation was successful or not equal 0 if failed.

SEE ALSO

rcadm_show(1M), rcadm_show_perf(1M), rcadm_statistics(1M), rcadm_set(1M), rcadm_unset(1M), rcadm_clone(1M), rcadm_revert(1M), rcadm_remove(1M), rcadm_commit(1M)

```
User Commands                                   rcadm -c statistics(1)


NAME
     rcadm -c statistics - Shows resource statistics


SYNOPSIS
     rcadm -c statistics


DESCRIPTION
     rcadm -c statistics shows the statistics about  the  current
     resource usage in the cpools managed by this VDCF instance.


EXAMPLES
     Following an example for rcadm -c statistics

        rcadm -c statistics


RETURN VALUES
     rcadm -c statistics returns 0 if the operation was  success-
     ful or not equal 0 if failed.


SEE ALSO
     rcadm_show(1M),     rcadm_show_perf(1M),      rcadm_set(1M),
     rcadm_unset(1M),      rcadm_clone(1M),     rcadm_revert(1M),
     rcadm_remove(1M), rcadm_commit(1M)




JomaSoft            Last change: 2008 Jun 1                   1
```

JomaSoft GmbH
Brauerstrasse 79a
9016 St. Gallen
Switzerland

NAME

rcadm -c unset - Used to unset/remove resource control properties

SYNOPSIS

```
rcadm -c unset
vserver=<vserver list>
props=<property list>
```

DESCRIPTION

rcadm -c unset is used to unset or remove a particular property from a vServer. This operations are always kept local in a uncommitted state. To transfer the settings to the particular vServer use the rcadm -c commit command. To inspect the actual and pending settings for a vServer use the rcadm -c show command.

EXAMPLES

Following an example for rcadm -c unset

```
rcadm -c unset vserver=s1024 props=locked,swap
```

RETURN VALUES

rcadm -c unset returns 0 if the operation was successful or not equal 0 if failed.

SEE ALSO

rcadm_show(1M), rcadm_show_perf(1M), rcadm_statistics(1M), rcadm_set(1M), rcadm_unset(1M), rcadm_clone(1M), rcadm_revert(1M), rcadm_remove(1M), rcadm_commit(1M)