
New Features and Usability Enhancements to RACADM Commands in iDRAC7

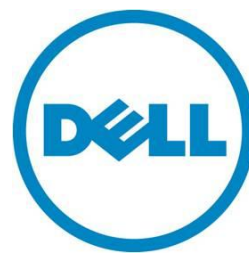
This Dell technical brief highlights the newest RACADM commands used with the iDRAC7 version 1.30.30

Rangarajan Varadan

Sumalatha Pagadala

Platform Software Engineers

iDRAC Team



Contents

Introduction	3
The Transformation	3
Simple keyword, context-sensitive commands.....	3
FQDD based syntax.....	3
Group and object syntax.....	3
New RACADM features	4
autocomplete	5
get and set.....	6
hwinventory.....	6
update	7
jobqueue.....	7
backup and restore	8
license	9
Configuring iDRAC, System, Lifecycle Controller, NIC, and BIOS using file operations.....	11
Summary.....	11
References.....	11

Tables

Table 1. iDRAC7 and RACADM features	4
Table 2. RACADM command options	6

This document is for informational purposes only and may contain typographical errors and technical inaccuracies. The content is provided as is, without express or implied warranties of any kind.

© 2013 Dell Inc. All rights reserved. Dell and its affiliates cannot be responsible for errors or omissions in typography or photography. Dell, the Dell logo, and PowerEdge are trademarks of Dell Inc. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

March 2013 | Version 1.0



Introduction

The Dell Remote Access Controller Admin utility (RACADM) is a command-line interface (CLI) used to locally or remotely configure the Integrated Dell Remote Access Controller (iDRAC) for Dell PowerEdge servers. The previous RACADM commands had long syntaxes, which made them difficult to use, and required training for new administrators. The newest version of RACADM, however, provides a wide range of simple, user-friendly command syntaxes.

Although open source CLI tools, such as WSMAN or IPMITool are available, Dell's intent for developing RACADM was to make it easy for you to configure the iDRAC. Along with the Dell PowerEdge 12th generation servers—the newest set of servers Dell offers—also available is a new set of RACADM features. With these new features, you now have a truly comprehensive and user-friendly CLI tool for configuring iDRAC7, System, Lifecycle Controller, NIC, and BIOS objects.

This technical brief primarily uses the secure shell (SSH)/Telnet interface to describe the new RACADM features, however the Local and Remote interfaces also support the new commands. For more information on RACADM, see the [RACADM Command Line Reference Guide for iDRAC7 1.30.30 and CMC4.3](#).

The Transformation

One of the primary improvements involved in making RACADM a more user-friendly tool was in standardizing the RACADM syntax by using:

- Simple keyword, context-sensitive commands
- Fully Qualified Descriptor (FQDD) based syntax
- Group and object syntax for configuring data

Simple keyword, context-sensitive commands

RACADM now uses the get syntax for getting (fetching) information, and a set syntax for setting (configuring) the iDRAC7 or a server. Note that there are exceptions for backward compatibility.

FQDD based syntax

FQDD based syntax allows each hardware component and software module on the iDRAC7 to be described with a unique FQDD. For example, iDRAC.embedded.1 is the FQDD for the iDRAC7.

Group and object syntax

Data for a particular module is categorized into an object and all the settings that can be configured or fetched under that module is an object group. An object might have sub-objects ranging from level one to higher. For example, in system.lcd.LCDUserString, system is the group, lcd is the object, and LCDUserString is the sub-object.



New RACADM features

In the iDRAC7 firmware releases spanning from 1.00.00 through 1.30.30, the features listed in Table 1 were introduced and supported by RACADM.

Table 1. iDRAC7 and RACADM features

iDRAC feature	Command syntax	Description	Supported interface
Auto complete	Supports most racadm subcommands	Displays available RACADM commands	telnet/ssh
Get	racadm get	Comprehensive settings get	telnet/ssh, local, remote
Set	racadm set	Comprehensive settings set	telnet/ssh, local, remote
Hardware inventory	racadm hwinventory	View and export server hardware inventory	telnet/ssh, local, remote
Update	racadm update	Update all server hardware including iDRAC and Lifecycle Controller	telnet/ssh, local, remote
Job control	racadm jobqueue	View, delete, create/schedule Lifecycle Controller jobs	telnet/ssh, local, remote
Backup and restore	racadm systemconfig	Backup and restore of server firmware and configuration	telnet/ssh, local, remote
Licensing	racadm license	License handling for iDRAC7	telnet/ssh, local, remote
Lifecycle Controller log	racadm lclog	Lifecycle Controller log view and export	telnet/ssh, local, remote
Inlet temperature	racadm inlettemphistory	FreshAir temperature history	telnet/ssh, local, remote
Event filter	racadm eventfilters (get/set)	Setting alerts and SNMP traps	telnet/ssh, local, remote



autocomplete

This feature enhances the usability of the RACADM CLI by displaying all the available RACADM commands in alphabetical order when you press Tab.

Using autocomplete for commands

To use this feature, press the Tab key to list all the RACADM subcommands in alphabetical order. If you type one or two letters and then press Tab, this feature automatically fills in the command.

For commands that begin with the same letter, autocomplete lists the possible commands that begin with that letter. For example, if you type `i` then press Tab, autocomplete lists the following:

```
ifconfig
inlettemphistory
```

To use autocomplete to execute a command, type part of the command that is unique, press Tab, and then press Enter. For example, to use autocomplete to execute the `ifconfig` command, type the first two letters of the command:

```
if
```

Press Tab to display the command:

```
ifconfig
```

And press Enter to execute the command.

Using autocomplete command options

The `racadm` shell now provides auto completion of subcommand options. If you know the command, but need to know the options available, you can enter the command and press Tab for autocomplete.

For example, see the following syntax for the `racadm license export` command:

```
racadm license export -f license_a.xml -u administrator -p Hello1$ -l
//10.194.181.221/DUPS -c idrac.embedded.1
```

Typically, if you didn't know what to enter after `export`, you could use the help command to find the answer. An easier way to find the answer is by using autocomplete. Simply press Tab after `export`, and autocomplete will enter the next necessary option for the command. For example, entering:

```
racadm>>license export <tab>
```

will display

```
racadm>>license export -f
```



In general, RACADM commands use the set of generic options listed in Table 2.

Table 2. RACADM command options

Option	Usage
-f	Usually followed by a filename
-l	Followed by a share path (NFS/CIFS)
-u	Followed by a username
-p	Followed by a password
-s	Used for fetching the status of an operation
-t	Used to specify a transaction ID
-i	Used when the command has indices (iDRAC users)

Note: Refer to command specific help for exact syntaxes and possible switches. Table 2 lists generic switch usage for most of the commands.

get and set

The get and set commands are based on the FQDD syntax. These new commands include the group and object categorization for settings. The group categories for get include:

- iDRAC
- System
- LifecycleController
- BIOS
- NIC

hwinventory

This feature allows you to view or export current server hardware inventory or shipped hardware inventory. To use this command, you must have login privileges. This command uses the following syntax:

```
racadm>> hwinventory
racadm>> hwinventory NIC
racadm>> hwinventory NIC.Embedded.1-1-2
racadm>> hwinventory export -f <filename> -u <username>
-p <password> -l <CIFS or NFS share>
```

For Local RACADM, use:

```
$ racadm hwinventory export -f <filename>
```



For Remote RACADM, use:

```
$ racadm -r <iDRAC-IP> -u <iDRAC username> -p <iDRAC Password> hwinventory  
export -f <filename>
```

Use cases

To display all server inventory, use:

```
$ racadm hwinventory
```

To view all the NIC devices on a managed server, use:

```
$ racadm hwinventory NIC
```

To display the standard hardware inventory verbose description for the FQDD NIC.Embedded.1-1-2, where NIC.Slot.5-2-2 is the FQDD, use:

```
$ racadm hwinventory NIC.Slot.5-2-2
```

To export the inventory to a remote CIFS share, use:

```
$ racadm hwinventory export -f Myinventory.xml -u admin -p mypass  
-l //192.162.31.41/share
```

To export the inventory to a remote NFS share, use:

```
$ racadm hwinventory export -f Myinventory.xml -u admin -p mypass  
-l 192.162.31.41:/share
```

update

This feature updates the firmware of various devices/modules on the server using the Dell Update Package (DUP) files (Windows DUPs, firmimg.d7, lc_repair_pack.usc and Personality_module.pm files only) either from a remote CIFS/NFS share or from a local file system using Local RACADM and Remote RACADM. This command uses the following syntax:

```
$ racadm update -f <updatefile>  
$ racadm update -f <updatefile> -l <remote CIFS/NFS share> -u <remote  
share username> -p <remote share password>
```

Use cases

For Remote RACADM, either CIFS/NFS use:

```
$ racadm update -f <update file> -u admin -p mypasswd -l  
//192.162.31.41/share [command for the]
```

For Local or Remote RACADM, use:

```
$ racadm update -f <update file> [command for the]
```

jobqueue

This feature allows you to view and delete job(s) in the current job queue. You must have the server profile export and import license to use this command.



Use cases

To view jobs in the current job queue, use:

```
$ racadm jobqueue view
```

To view jobs in the current job queue and display the specific job ID, use:

```
$ racadm jobqueue view -i <JobID>
```

To delete all possible jobs from the current job queue, use:

```
$ racadm jobqueue delete -all
```

To delete a specific job from the current job queue, use:

```
$ racadm jobqueue delete -i <JobID>
```

To create a job for the provided FQDD and add to the job queue, use:

```
$ racadm jobqueue create <fqdd> [-r <reboot type>] [-s <start time>] [-e  
wait time]
```

```
$ racadm jobqueue create NIC.Integrated.1-1 -r pwr cycle -s TIME_NOW -e  
20120501100000
```

backup and restore

Use the `racadm systemconfig` command to backup or restore server firmware and configuration. You must have the server profile export and import license to use this command. Backup is a licensed (Enterprise) feature, but restore is a non-licensed feature.

If Lifecycle Controller is disabled, restore will not work. If Collect System Inventory on Reboot (CSIOR) is disabled, the system inventory may not be current during a backup operation. An appropriate warning message is reported.

The job ID is displayed when the backup or restore is successful. `<systemconfig command type>` is either backup or restore.

```
$ racadm systemconfig <systemconfig command type>
```

The syntax for backup is:

```
$ racadm systemconfig backup -f <filename> <target> [-n passphrase] [-l  
<location> -u <username> -p <password>] [--vFlash]
```

The syntax for restore is:

```
$ racadm systemconfig restore -f <filename> <target> [-n passphrase] [--  
nopreserve] [-l <location> -u <username> -p <password>]  
[--vFlash]
```

Use cases

To back up a system to a CIFS share and encrypt the data, use:

```
$ racadm systemconfig backup -f image.img -l //192.168.2.140/share  
-u admin -p passwd -n encryptpasswd123 []
```



To back up a system to a NFS share and encrypt the data, use:

```
$ racadm systemconfig backup -f image.img -l 192.168.2.140:/share
-u admin -p passwd -n encryptpasswd123
```

To back up a system to a vFlash SD, use:

```
$ racadm systemconfig backup -vFlash
```

To restore a system from a vFlash SD and clear the Virtual Disks (VD) configurations, use:

```
$ racadm systemconfig restore -vFlash -nopreserve
```

To restore system from an NFS share without clearing the VD configurations, use:

```
$ racadm systemconfig restore -f image.img -l 192.168.2.140:/share
-u admin -p passwd
```

To create a backup file in a subfolder within the CIFS shared folder, use:

```
$ racadm systemconfig backup -f rts/Backup.img -l
//10.94.161.200/CIFSshare -u username -p password
```

To view the status of an ongoing systemconfig operation with the job ID 100, use:

```
$ racadm jobqueue view -i 100
```

license

This command is used to manage hardware licenses. The name of the <license-file> should be less than 56 characters. During remote file share, SSH/Telnet supports import and export options. You must have iDRAC administrator privileges to use the import, export, delete, and replace commands. You do not need administrator privileges to use the view command. This command uses the following syntax:

```
$ racadm license view
$ racadm license view [-c <component>]
$ racadm license import [-f <licensefile>] -l <location> -c <component>
$ racadm license export [-f <license file>] -e <ID> -c <component>
$ racadm license delete -e <ID> -l <location> [-f] -c <component> -t
<transaction ID>
$ racadm license delete -t <transaction ID>
$ racadm license delete -e <entitlement ID>
$ racadm license delete -c <component>
$ racadm license replace -t 1
$ racadm license replace -u <username>-p <password> -f <license file name>
-l <NFS/CIFS share> -t <transaction ID>
```

Use cases

To view all license information on the system, use:

```
$ racadm license view
```



To import a new license to a specific device in a known location, use:

```
$ racadm license import -f license.xml -l //shareip/sharename  
-u <share user> -p <share user password> -c idrac.embedded.1
```

To import a license from a CIFS share to a device, in this case embedded iDRAC, use:

```
$ racadm license import -u admin -p passwd -f License.xml-l  
//192.168.2.140/licshare -c idrac.embedded.1
```

To import a license from the local file system using Local RACADM, use:

```
$ racadm license import -f License.xml -c idrac.embedded.1
```

To import a license from the local filesystem using Remote RACADM, use:

```
$ racadm license import -f C:\Mylicdir\License.xml  
-c idrac.embedded.1
```

To export a license file (for Remote RACADM, if filename is not specified, the file(s) are exported to the directory where RACADM is running), use:

```
$ racadm license export -f <filename> -l <share location>  
-u <share username> -p <share password> -c iDRAC.Embedded.1
```

To delete a license, use:

```
$ racadm license delete -e B9865F23455DC458
```

To delete all licenses installed on a particular device, use:

```
$ racadm license delete -c idrac.embedded.1
```

To delete licenses on a particular device, in this case embedded iDRAC, use:

```
$ racadm license delete -c idrac.embedded.1
```

To delete a license using entitlement ID, in this case xYZabcdefg, use:

```
$ racadm license delete -e xYZabcdefg
```

To delete a license using transaction ID, in this case 2, use:

```
$ racadm license delete -t 2
```

To replace a license on a device with a license file on an NFS share using transaction ID, in this case transaction 27, use:

```
$ racadm license replace -f License.xml -l 192.168.2.140  
:/licshare-t 27
```

To replace a license on a device with a license file on a CIFS share using transaction ID, in this case transaction 27, use:

```
$ racadm license replace -u admin -p passwd -f License.xml-l  
//192.168.2.140/licshare -t 27
```



Configuring iDRAC, System, Lifecycle Controller, NIC, and BIOS using file operations

The legacy method of using RACADM to configure the iDRAC is commonly called the file operations:

```
$ racadm getconfig -f <filename>
```

The output file for this command contains several iDRAC settings. The same file is used for configuring a single or a set of iDRACs using the following command:

```
$ racadm config -f <filename>
```

In the newest RACADM version, file operations has been enhanced by:

- Using get and set in place of config:
 - `racadm get -f <filename>`
 - `racadm set -f <filename>`
- Server profile now includes Eventfilters, NIC objects, RAID Objects, BIOS, System, Lifecycle Controller and iDRAC objects using:

```
$ racadm get -f <file.xml> -t xml/csv
```

and

```
$ racadm set -f <file.xml> -t xml/csv
```
- The `-t` option creates a configuration file in XML or CSV format based on the user input, and the file created can be exported to:
 - Remote file share using the `-l` option for a CIFS or NFS share destination
 - Local file system of a management station (Remote RACADM) or server (Local RACADM)

Summary

RACADM has transformed to a standardized, user-friendly interface. It is becoming a comprehensive CLI as Dell continues to provide new and unique features found only in the Dell PowerEdge server line with iDRAC7 and Lifecycle Controller.

In the future, Dell will continue improving the RACADM CLI to make it compatible with all Dell enterprise products, and further improve user experience in the overall configuration of server deployment.

References

For more information on iDRAC7 version 1.30.30, visit [Support for Integrated Dell Remote Access Controller 7 Version 1.30.30](#) on [Dell.com](#).

For additional info on iDRAC, visit [Dell Remote Access Controller](#) on [Dell.com](#).

