

Configuring BIOS and NIC Using RACADM



Notes, Cautions, and Warnings



NOTE: A NOTE indicates important information that helps you make better use of your computer.



CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.



WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

© 2013 Dell Inc.

Trademarks used in this text: Dell™, the Dell logo, Dell Boomi™, Dell Precision™, OptiPlex™, Latitude™, PowerEdge™, PowerVault™, PowerConnect™, OpenManage™, EqualLogic™, Compellent™, KACE™, FlexAddress™, Force10™ and Vostro™ are trademarks of Dell Inc. Intel®, Pentium®, Xeon®, Core® and Celeron® are registered trademarks of Intel Corporation in the U.S. and other countries. AMD® is a registered trademark and AMD Opteron™, AMD Phenom™ and AMD Sempron™ are trademarks of Advanced Micro Devices, Inc. Microsoft®, Windows®, Windows Server®, Internet Explorer®, MS-DOS®, Windows Vista® and Active Directory® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat® and Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc. in the United States and/or other countries. Novell® and SUSE® are registered trademarks of Novell Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation and/or its affiliates. Citrix®, Xen®, XenServer® and XenMotion® are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. VMware®, vMotion®, vCenter®, vCenter SRM™ and vSphere® are registered trademarks or trademarks of VMware, Inc. in the United States or other countries. IBM® is a registered trademark of International Business Machines Corporation.

2013 - 07

Rev. A00

Introduction

Today you can update BIOS and firmware out-of-band using the WSMAN protocol and DTK but this does not support the most common interfaces used with iDRAC7 (Web interface and Command Line Interface). From iDRAC7 1.31.30 release onwards, you can use the new and simplified command-line syntax to update the BIOS and NIC using RACADM CLI. A server reboot is required. It supports full server configuration and updates.

The following two features were also introduced:

- `cd` command to navigate between groups and attributes within the RACADM shell.
- Use the `<TAB>` key to automatically complete the command.

This document provides a brief introduction on the BIOS and NIC attributes that you can configure using the RACADM interface. It also provides information about `cd` based navigation support for groups and attributes, and `TAB` based auto complete feature.

Configuring BIOS Using RACADM

The following sequence of commands provides the steps to get and set the BIOS attribute using examples.



NOTE:

- Make sure that Lifecycle Controller and Collect System Inventory On Reboot (CSIOR) is enabled before performing the configuration.
- BIOS settings take effect only after server has rebooted and CSIOR has run.
- Attribute value names are case-sensitive.
- Few attributes mentioned in this document are platform specific and may depend on the BIOS and Lifecycle Controller versions, type of the NIC card, system configuration, and so on.

To view and configure the BIOS attributes using RACADM:

1. Get the list of attributes for a specific group (for example, MemSettings):

```
$ racadm get BIOS. MemSettings
[Key=BIOS.Setup.1-1#MemSettings]
#MemOpMode=OptimizerMode
MemTest=Disabled
#MemVolt=AutoVolt
NodeInterleave=Disabled
#SysMemSize=2.0 GB
#SysMemSpeed=1333 MHz
#SysMemType=ECC DDR3
#SysMemVolt=1.5V
#VideoMem=16 MB
```

2. Query the value for a specific attribute (for example, MemTest in the MemSettings group):

```
$ racadm get BIOS.MemSettings.Memtest
[Key=BIOS.Setup.1-1#MemSettings]
MemTest=Disabled
```

3. Modify the value for an attribute (for example, enable MemTest in the MemSettings group):

```
$ racadm set bios.memsettings.memtest Enabled
[Key=BIOS.Setup.1-1#MemSettings]
RACl017: Successfully modified the object value and the change is in
pending state.
To apply modified value, create a configuration job and reboot the system.
To create the commit and reboot jobs, use "jobqueue" command. For more
information about the "jobqueue" command, see RACADM help.
```

4. Query the value for the recently modified attribute:

```
$ racadm get bios.memsettings.memtest
[Key=BIOS.Setup.1-1#MemSettings]
MemTest=Disabled (Pending Value=Enabled)
```
5. Create the commit and reboot jobs using jobqueue create command and apply the modified BIOS attribute value:

```
$ racadm jobqueue create BIOS.Setup.1-1 -r pwr cycle -s TIME_NOW -e TIME_NA
RAC1024: Successfully scheduled a job.
Verify the job status using "racadm jobqueue view -i JID_XXXXX" command.
Commit JID = JID_497398488366
Reboot JID = RID_497398488646
```
6. (Optional) A commit job can be created without creating a reboot job. In such a case, you can reboot the OS using the racadm serveraction command.

```
$ racadm jobqueue create BIOS.Setup.1-1
RAC1024: Successfully scheduled a job.
Verify the job status using "racadm jobqueue view -i JID_XXXXX" command.
Commit JID = JID_497398488366
```
7. Check the job status of all the jobs or a specific job using jobqueue view command:

```
$ racadm jobqueue view -i JID_497398488366

or

$ racadm jobqueue view

-----JOB QUEUE-----
[Job ID=JID_497398488366]
Job Name=Configure: BIOS.Setup.1-1
Status=Completed
Start Time=[Now]
Expiration Time=[Not Applicable]
Message=[PR19: Job completed successfully.]
-----
[Job ID=RID_497398488646]
Job Name=Reboot: Power cycle
Status=Reboot Completed
Start Time=[Now]
Expiration Time=[Not Applicable]
Message=[RED030: Reboot is complete.]
```

Miscellaneous Commands Under BIOS Group

1. View the complete list of supported BIOS groups:

```
$ racadm get bios
```
2. To view information about a related group or an attribute:

```
$ racadm help bios.memsettings

or

$ racadm help bios.memsettings.memtest
```
3. Change boot order settings (for example, UEFI):
 - a) Make sure that the current BootMode under the Biosbootsettings group is set to 'Uefi' for UEFI boot sequence changes.

```
$ racadm get bios.biosbootsettings.BootMode
[Key=BIOS.Setup.1-1#biosbootsettings]
BootMode=Uefi
```
 - b) Get the current UEFI Boot order:

```
$ racadm get bios.biosbootsettings.UefiBootSeq
[Key=BIOS.Setup.1-1#biosbootsettings]
```

- ```
UefiBootSeq=NIC.Integrated.1-4-1,NIC.Integrated.1-2-1,NIC.Integrated.1-3-1,NIC.Integrated.1-1-1
```
- c) Change the UEFI boot order:
- ```
$ racadm set bios.biosbootsettings.UefiBootSeq NIC.Integrated.1-4-1,NIC.Integrated.1-2-1,NIC.Integrated.1-1-1,NIC.Integrated.1-3-1
[Key=BIOS.Setup.1-1#biosbootsettings]
RAC1017: Successfully modified the object value and the change is in pending state.
```
- To apply modified value, create a configuration job and reboot the system. To create the commit and reboot jobs, use "jobqueue" command. For more information about the "jobqueue" command, see RACADM help.
- d) Verify the modified attribute value:
- ```
$ racadm get bios.biosbootsettings.UefiBootSeq
[Key=BIOS.Setup.1-1#biosbootsettings]
UefiBootSeq=NIC.Integrated.1-1-1,NIC.Integrated.1-2-1,NIC.Integrated.1-3-1,NIC.Integrated.1-4-1
(Pending Value=NIC.Integrated.1-4-1,NIC.Integrated.1-2-1,NIC.Integrated.1-1-1,NIC.Integrated.1-3-1)
```
- e) Create the BIOS configuration job as explained in the previous section.

## Configuring NIC Using RACADM


Use the NIC instance associated with the corresponding NIC key to query the attribute or configure the settings. Each NIC key has the following format:

Key = NIC.<Locator>.<Device Number>—<Port Number>[—<Partition Number>]#GroupName

where, Locator value = Integrated, Slot, Mezzanine, or Embedded.

For example:


```
$racadm get nic.nicconfig
NIC.nicconfig.1 [Key=NIC.Integrated.1-1#NICConfig]
NIC.nicconfig.2 [Key=NIC.Integrated.1-2#NICConfig]
NIC.nicconfig.3 [Key=NIC.Integrated.1-3#NICConfig]
NIC.nicconfig.4 [Key=NIC.Integrated.1-4#NICConfig]
```

 **NOTE:** The co-relation between the NIC instance and the corresponding key varies from system to system depending on the system configuration.

The following example provides the steps to configure the LegacyBootProto NIC attribute in the NICConfig group for NIC Device 1, Port 3.

1. Get the current value of the NIC Config group:

```
$ racadm get nic.nicconfig.3
[Key=NIC.Integrated.1-3#NICConfig]
LegacyBootProto=NONE
LnkSpeed=AutoNeg
VlanId=1
VlanMode=Disabled
WakeOnLan=Disabled
```

 **NOTE:** To view the list of NIC groups, use the racadm get nic command.

2. Configure the LegacyBootProto to PXE:

```
$ racadm set nic.nicconfig.3.legacybootproto PXE
[Key=NIC.Integrated.1-3#NICConfig]
RAC1017: Successfully modified the object value and the change is in pending state.
```

To apply modified value, create a configuration job and reboot the system. To create the commit and reboot jobs, use "jobqueue" command. For more information about the "jobqueue" command, see RACADM help.

3. Query the modified attribute value. The new value is in pending state.  

```
$racadm get nic.nicconfig.3.legacybootproto
[Key=NIC.Integrated.1-3#NICConfig]
LegacyBootProto=NONE (Pending Value=PXE)
```
4. Create a NIC configuration job for NIC Device 1, Port 3 to commit and apply the configured value:  

```
$ racadm jobqueue create NIC.Integrated.1-3
RAC1024: Successfully scheduled a job.
Verify the job status using "racadm jobqueue view -i JID_XXXXX" command.
Commit JID = JID_637820565389
```
5. Reboot the host operating system and make sure that CSIOR has run for the changes to be applied successfully.

## Navigating Between Group Or Attributes Using cd Command

You can navigate between groups and attributes using the cd command in the RACADM shell. Using cd, you can change to a group, an attribute within the group, or move back into the parent group. As an example, this section provides the steps to configure a BIOS attribute using the cd based navigation within the RACADM shell.

1. Open a SSH/Telnet session, at the SMCLP shell, run the racadm command to enter the RACADM shell:  

```
login as: root
root@10.94.161.120's password:
WARNING: Default password is configured. Dell highly recommends changing
user root's password immediately
/admin1-> racadm
```
2. Navigate to BIOS device type:  

```
racadm>>cd BIOS
racadm/BIOS>
```
3. Navigate to MemSettings group:  

```
racadm/BIOS>cd MemSettings
racadm/MemSettings>
```
4. Get the list of attributes under BIOS.Memsettings:  

```
racadm/MemSettings>get
racadm get BIOS.MemSettings
[Key=BIOS.Setup.1-1#MemSettings]
#MemOpMode=OptimizerMode
MemTest=Enabled
#MemVolt=AutoVolt
NodeInterleave=Disabled
#SysMemSize=2.0 GB
#SysMemSpeed=1067 MHz
#SysMemType=ECC DDR3
#SysMemVolt=1.35V
#VideoMem=16 MB
```
5. Set attribute value for MemTest:  

```
racadm/MemSettings>set MemTest Disabled
racadm set BIOS.MemSettings.MemTest Disabled
RAC1017: Successfully modified the object value and the change is in
pending state.
To apply modified value, create a configuration job and reboot the system.
To create the commit and reboot jobs, use "jobqueue" command. For more
information about the "jobqueue" command, see RACADM help.
```
6. Navigate back to the parent directory:  

```
racadm/MemSettings> cd ..
racadm/BIOS> cd ..
racadm>>
```

7. Quit from the RACADM shell:

```
racadm>> quit
```

## Using TAB Key to Autocomplete a RACADM Command

The RACADM SSH interface supports tab-based auto completion. If there are multiple options matching the text after pressing the <TAB> key, all the options are displayed. Auto-completion helps you to complete a command or quickly know the available options.

For example, to retrieve BIOS and NIC attributes:

```
racadm>>get BIOS <TAB>
BiosBootSettings
IntegratedDevices
MemSettings
MiscSettings
OneTimeBoot
ProcSettings
SataSettings
SerialCommSettings
SlotDisablement
SysInformation
SysProfileSettings
SysSecurity

racadm>>get BIOS MiscSettings <TAB>
AssetTag
Characterization
ErrPrompt
NumLock
ReportKbdErr
SystemUefiShell

racadm>>get BIOS MiscSettings A <TAB>

racadm>>get BIOS MiscSettings AssetTag <TAB>
racadm get BIOS.MiscSettings.AssetTag.
[Key=BIOS.Setup.1-1#MiscSettings]
AssetTag=XYZ

racadm>>get NIC <TAB>
DCBSettings
DeviceLevelConfig
FCOECapabilities
FCOEConfiguration
FCOEGenParams
FrmwImgMenu
GlobalBandwidthAllocation
IscsiFirstTgtParams
IscsiGenParams
IscsiInitiatorParams
IscsiSecondaryDeviceParams
IscsiSecondTgtParams
NICConfig
NICPartitioningConfig
VndrConfigGroup

racadm>>get NIC DCBSettings <TAB>
<INT:index>

racadm>>get NIC DCBSettings 1 <TAB>
CongestionNotification
DCBExchangeProtocol
```

```
EnhancedTransmissionSelection
PriorityFlowControl
```

```
racadm>>get NIC DCBSettings 1 Cong <TAB>
```

```
racadm>>get NIC DCBSettings 1 CongestionNotification <TAB>
racadm get NIC.DCBSettings.1.CongestionNotification.
[Key=NIC.Integrated.1-1-1#DCBSettings]
CongestionNotification=Available
```

## Conclusion

The iDRAC7 RACADM CLI enables you to configure hardware and systems settings (including BIOS, NIC, System, Lifecycle Controller, and iDRAC). These configurations can be scripted and run either in-band or through the remote console, thereby providing a quick and easy method to deploy and manage servers at Data Centers. Though this document provides information about cd based navigation and autocomplete features for BIOS and NIC device types, you can use these features for iDRAC, System, and Lifecycle Controller devices.

## Learn More

For more information on RACADM sub commands and objects, see the [RACADM Command Line Reference Guide](#) .

For details on iDRAC7 with Lifecycle Controller, see the iDRAC page on Dell Tech Center:

<http://www.delltechcenter.com/idrac> .